SAP CPS Administration Guide

Version M28.49-54820

Table of Contents

1. Welcome 1
    - Use 1
        - Introduction 1
    - See Also 2

2. Release Notes 3
    - Browser Versions 3
    - Reverting from Chargeable back to Free of Charge Version is NOT Supported 3
    - Run As Fields and Special Characters in Passwords 3
    - Redwood Platform and Oracle Prerequisites 3
    - Tablespace Usage - Upgrading from Version 7 4
    - Killing RedwoodScript Jobs 4
    - FileInUseCommand Process Server Parameter Changed Default Value on Unix 4
    - SAP Application Return Codes 5
    - Dates are Lenient 5
    - SQL Server Deadlocks 5
    - Job Chain Status 5
    - Preconditions 6
    - Job Run Order Change 6
    - OnChange Trigger 6
    - Firefox 3.6 and Java Plugins 6
    - Debian-based Systems and Java Networking 7
    - Import Jobs and Clusters 7
    - SQL Server JDBC Driver Settings 7
    - SAP Interception 7
    - JVM Time Zone Settings 8
    - HTTP Worker Threads 8
    - Oracle Globalization Support 8
    - Clients Require Time Zone Data 8
    - File Event Move Directories 8
    - Default Pagesize on DB2 is insufficient 8
    - Queues, Time Windows and Time Zones 9
    - Platform Agent Network Timeout 9
    - Fail-over Timeouts 9
    - File:// links and Mozilla Firefox 9
    - Fail-over Timeouts 10
    - Time Windows and the Custom Library 10
    - Shell 10
    - Transport Files 10
    - Changing the SAP System of a Job with Default Queue 10
    - SAP Notes 10
    - Upgrade Issues 10
        - Upgrading from 26 to 28 10
        - SAP Job Definitions 11
        - SAP Job Names 11
        - Events and Job Files 11
        - Application Renames 11
    - Platform Agents on Linux 11
    - Platform Agents on Tru64 12
    - JCo Dependency Issues 12
        - On UNIX and GNU/Linux 12
        - Windows 12

3. New Features 13

Import Jobs and Clusters ... 13
Audited Objects have and Audit Trail tab ... 13
Improved Security with sudo ... 13
Platform Agent Network Timeout ... 13
Fail-over Timeouts ... 13
Permalink Module ... 13
Links in Comment Fields ... 13
SAP SNC Support ... 14
Advanced Auditing Rules ... 14
Advanced Search for all Objects ... 14
Easy Retrieval of Process Server Logs ... 14
Controlling Communication Channels with SAP PI ... 14
New Features in M26 ... 14
SAP Applications API Integration ... 14
Mail Job Definition Type ... 14
Period Functions ... 15
Reports ... 15
Credentials ... 15
Dashboard ... 15
Library and Triggers ... 15
Isolation groups ... 15
4. SAP CPS ... 16
Use ... 16
Implementation Considerations ... 16
Integration ... 17
Features ... 17
Activities ... 17
More Information ... 18
5. Architecture ... 19
Technical System Landscape ... 19
Use ... 19
Integration ... 19
See Also ... 20
Central Scheduling Concepts ... 20
Use ... 20
Jobs, Job Chains, Job History ... 20
Queues and Process Servers ... 21
Distribute Workload ... 23
SAP Systems ... 25
Events and Event Definitions ... 27
Scheduling ... 27
Applications ... 30
6. Getting Started - Using SAP CPS ... 31
Use ... 31
Logging on ... 31
Use ... 31
Version Information and Help ... 31
Prerequisites ... 32
Procedure ... 32
Navigation ... 32
Use ... 32
The Title bar and Tool bar ... 34
Actions ... 35
Restrictions on Actions ... 36
Navigating with Screen Readers ... 36

Performing Basic Actions 37
Use 37
Procedure 40
Locating Objects Quickly 41
Use 41
Procedure 42
Result 42
Example 42
Advanced Object Search 42
Advanced Actions 44
Use 44
Editing Objects in XML 44
7. Administering the Scheduling Landscape 46
Use 46
Checking Your License 47
Use 47
Special Restrictions 49
See Also 50
Different Versions 50
Configuring the Central and Satellite Systems 53
Use 53
Understanding the Different XBP Interface Releases 54
Enhanced SAP Interfaces with Redwood Transports 57
SAP Communication Channels with PI/XI 60
Configuring an SAP System for Central Process Scheduling 61
Connecting to SAP Systems 64
Managing Roles and Users 98
Use 98
Isolating Users and Objects 99
Managing Users and Roles in External Systems 100
Managing Users and Roles in SAP CPS 107
Creating Isolation Groups and Assigning them to Users 107
Assign Users from External Platforms to Jobs with Credentials 108
Privileges 114
Auditing Object Changes 174
Use 174
Procedure 177
Example 178
Generating Reports 178
Use 178
Procedure 180
Example 181
Basic Notification and Alerting with the Mail Module 181
Use 181
See Also 182
Configuring Mail and Sending Emails 183
Customizing Emails 186
Advanced Notification and Alerting with the Active Monitoring Module 187
Use 187
Triggering Alerts 189
Defining Escalation Routes 205
Email Alert Gateways 209
Setting User Parameters in the Options dialog 214
Use 214
Locale Format 215

The Registry ... 215
Use ... 215
Procedure ... 216
Scaling the Scheduling Resources on the Central System ... 217
Use ... 217
Using Process Servers ... 218
Using Queues ... 231
Prevent Simultaneous Execution of Processes ... 242
Creating Event-Driven Schedules with Events ... 244
Tuning SAP CPS ... 253
Tuning Memory Usage/Garbage Collection ... 255
Balancing the Load ... 257
High Availability ... 262
Organizing Processes ... 272
Use ... 272
See Also ... 274
Creating Applications ... 274
Managing Applications with RedwoodScript ... 275
Importing and Exporting SAP CPS Objects ... 275
Use ... 275
Procedure ... 279
Example ... 280
Managing Job Output Formats ... 282
Use ... 282
Procedure ... 284
Example ... 285
See Also ... 285
Regular Activities for SAP CPS ... 285
Use ... 285
Backup and Recovery of SAP CPS ... 287
Use ... 287
Architecture ... 287
Backup ... 288
See Also ... 289
Database Backup ... 289
Restore Data from Backup ... 291
8. Automating Processes Across the System Landscape ... 293
Creating and Editing Job Definitions ... 293
Use ... 293
Procedure ... 296
See Also ... 297
Setting Job Definition Properties ... 297
Redwood Job Definition Types ... 319
Creating Job Chains ... 324
Use ... 324
Procedure ... 328
Job Chain Steps ... 329
Job Chain Jobs ... 330
Creating Job Chains with Sequential Jobs ... 332
Creating Job Chains with Parameter Handling ... 334
Creating Job Chains with Parallel Jobs ... 336
Creating Job Chains with Job Dependencies ... 337
Creating Job Chains for Multiple SAP Systems ... 340
Precondition Functions ... 342
Redwood Expression Language ... 344

Submitting Jobs and Job Chains ... 352
Use ... 352
See Also ... 352
Submitting a Job or Job Chain ... 352
Scheduling with Times and Dates ... 355
See Also ... 355
Rescheduling Jobs Automatically ... 356
Defining Execution Times with Time Windows ... 363
Time Zones ... 374
Importing Objects from SAP Systems ... 376
Use ... 376
See Also ... 377
Importing ABAP Programs ... 377
Importing ABAP Program Parameter Definitions ... 378
Importing Archive Objects ... 379
Importing Batch Events ... 380
Importing Calendars ... 381
Importing CCMS Jobs ... 381
Importing Communication Channels via PI/XI ... 383
Importing External Commands ... 384
Importing Java Job Definitions ... 385
Importing and Running Mass Activities ... 386
Importing Output Devices ... 387
Importing and Running Process Chains ... 388
Importing Recipients ... 391
Importing Variants ... 392
Creating and Modifying Variants ... 393
Maintaining Variants Manually ... 395
Maintaining Variants Automatically ... 396
Maintaining Variants Automatically (Advanced) ... 399
Using Temporary Variants ... 400
Redwood Expression Language ... 403
Use ... 403
Operators ... 403
Escape Character ... 404
Predefined functions ... 404
See Also ... 411
RedwoodScript ... 411
Use ... 411
Scripting in the Shell ... 417
Triggering Custom Code ... 418
Use ... 418
Using Triggers ... 418
Using Actions ... 426
Executing Your Own Classes from a Library ... 428
9. Monitoring Automated Processes ... 435
Monitoring and Managing Jobs ... 435
Use ... 435
Procedure ... 437
See Also ... 438
Available Columns in the Job Monitor ... 438
Job States ... 439
Filtering Jobs ... 443
Advanced Job Search ... 445
Viewing Jobs in a Calendar View ... 447

SAP CPS job Processing ... 448
Monitoring Events ... 449
Use ... 449
Procedure ... 450
Filtering Events ... 450
Monitoring System Performance ... 452
Use ... 452
Procedure ... 453
Showing Monitors as Graphs in the Dashboard ... 454
Monitoring SAP Systems ... 456
Use ... 456
Interacting with Operator Messages ... 456
Use ... 456
Procedure ... 458
Example ... 458
Receiving Operator Messages on your Desktop ... 460
Creating Advanced Reply Expressions ... 462
Integrating SAP CPS into your SNMP Landscape ... 463
Use ... 463
Example ... 464
See Also ... 464
Sending an SNMP Trap ... 465
Integrating SAP CPS into Third-Party Applications ... 466
Use ... 466
Prerequisites ... 466
list ... 466
show ... 467
edit ... 467
create ... 468
submit ... 468
help ... 468
10. Scheduling Processes Outside of the SAP System Landscape ... 469
Use ... 469
See Also ... 469
Using Platform Job Definition Types ... 469
Use ... 469
See Also ... 471
Unix Job Definition Types ... 471
Microsoft Windows Job Definition Types ... 476
HP OpenVMS Job Definition Type ... 480
The Environment of Platform Jobs ... 482
Using External Job Definition Types ... 483
Use ... 483
See Also ... 484
IBM z/OS Job Definition Types ... 484
IBM AS/400 Job Definition Types ... 490
Creating Platform Process Servers ... 495
Use ... 495
Agent Connection ... 496
Processing Platform Jobs ... 496
UNIX Process Servers ... 497
Creating a Microsoft Windows Process Server ... 505
Creating HP OpenVMS Process Servers ... 509
IBM z/OS Process Servers ... 518
Configuring Platform Agents ... 521

Command Line System Tools 536
System Tools and Platform Agent Logging 549
Creating AS/400 Process Servers 551
Use 551
Prerequisites 552
Procedure 552
Files on AS/400 Raise Events 554
11. Reference 555
SAP RFC User Privileges 555
Use 555
SAP Authorizations for XBP and BW 555
AAAB - Cross-application Authorization Objects 555
BC_A - Basis: Administration 556
SAP Authorizations for BW Process Chains 558
S_DEVELOP - ABAP Workbench 559
AAAB - Cross-application Authorization Objects 559
SAP Authorizations for Industry Solutions (ISU) 559
SAP Authorizations for SAP Applications 559
Default System Job Definitions 560
Use 574
System_Alert_Send 561
System_ArchiveAuditObjects 562
System_ArchiveEvents 562
System_Backup 562
System_CreateMonitorTree 563
System_DeleteJobFiles 563
System_Derby_Backup 563
System_DynamicTrace 564
System_ExecuteExternalJob 564
System_ExplainJobStatus 564
System_Export_Application 565
System_Export_RuleSet 565
System_ForceJobStatusUnknown 565
System_GetSupportFiles 566
System_Get_Tutorials_List 566
System_Ignored_Alert_Reporter 566
System_Import_Archive 566
System_Import_JarFile 567
System_Info 567
System_Isolation_Deisolation 567
System_Isolation_SetUser 567
System_Mail_Configure 568
System_Mail_Send 568
System_OperatorMessage 568
System_ProcessKeepClauses 569
System_RaiseClearEvent 569
System_ReconfigureLogging 569
System_RemoveDeletedJobFiles 569
System_RemoveOldJobs 570
System_RemoveRepliedMessages 570
System_ReportRun 571
System_Restore_Object 571
System_Restore_Point_In_Time 571
System_SetDefaultLanguage 572
System_SetSystemDataRootDirectory 572

System_Sleep ... 572
System_SMS_Configure ... 572
System_SMS_Send ... 573
System_SNMP_SendTrap ... 573
System_UpdateJobStatistics ... 573
System_Upgrade ... 574
System_Version_Info ... 574
Default SAP Job Definitions ... 574
Use ... 621
SAP_AbapRun ... 577
SAP_AbapRunPrint ... 579
SAP_AbapRunPrintExt ... 581
SAP_BW_GenerateInfoPackageGroup ... 583
SAP_BW_ImportInfoPackageGroups ... 583
SAP_BW_ImportInfoPackages ... 584
SAP_BW_ImportProcessChainDefinition ... 584
SAP_BW_ImportProcessChains ... 584
SAP_BW_InfopackageRun ... 585
SAP_BW_MaintainProcessChainDefinition ... 585
SAP_BW_Process ... 586
SAP_BW_ProcessChain ... 586
SAP_BW_RestartProcess ... 587
SAP_BW_RestartProcessChain ... 587
SAP_BW_ProcessChainRun ... 588
SAP_CcmsJob ... 588
SAP_CcmsJobCheckStatus ... 589
SAP_CcmsJobGetApplicationLog ... 590
SAP_CcmsJobGetApplicationLogHandles ... 590
SAP_CcmsJobGetLog ... 591
SAP_CcmsJobGetSpool list ... 591
SAP_CcmsJobGetStatistics ... 591
SAP_CcmsJobModifyHeader ... 592
SAP_CcmsJobGetStatus ... 592
SAP_ClosingCockpit_AbapRun ... 593
SAP_ClosingCockpit_ManualTask ... 595
SAP_CopyVariant ... 596
SAP_DeleteCcmsMonitors ... 596
SAP_DeleteVariant ... 597
SAP_ExternalCommandRun ... 597
SAP_ExternalProgramRun ... 598
SAP_GenerateCcmsJobDefinition ... 599
SAP_GetBatchResources ... 600
SAP_GetEventHistory ... 600
SAP_GetInstalledComponents ... 600
SAP_GetXmiLog ... 601
SAP_ImportAbapParameterDefinition ... 601
SAP_ImportAbapPrograms ... 601
SAP_ImportApplicationServers ... 602
SAP_ImportArchiveObjects ... 602
SAP_ImportBatchEvents ... 603
SAP_ImportCalendars ... 603
SAP_ImportExternalCommands ... 603
SAP_ImportCcmsJobs ... 604
SAP_ImportCcmsMonitors ... 605
SAP_ImportOutputDevices ... 605

SAP_ImportRecipients 606
SAP_ImportVariantDefinition 606
SAP_ImportVariants 607
SAP_Info 607
SAP_InterceptedJob 607
SAP_JXBP_GenerateJobDefinition 608
SAP_JXBP_ImportJobDefinitions 608
SAP_MA_CopyParameters 609
SAP_MA_CreateParameters 609
SAP_MA_DeleteDunningProposal 610
SAP_MA_ImportMassActivities 610
SAP_MA_ImportMassActivity 611
SAP_MA_MassActivityRun 611
SAP_MA_MassActivityTemplateRun 612
SAP_MA_ResumeMassActivity 612
SAP_MA_ShowMassActivity 612
SAP_MA_StopMassActivity 613
SAP_MaintainXbpOptions 613
SAP_PI_GetChannelStatus 614
Parameters 614
SAP_PI_ImportChannels 614
SAP_PI_StartChannel 615
SAP_PI_StopChannel 615
SAP_RaiseEvent 615
SAP_ReloadApplicationLinks 616
SAP_ReleaseInterceptedJobs 616
SAP_RunTemplateJob 616
SAP_SetXmiAuditLevel 617
SAP_SolutionManager_AbapRun 617
SAP_StoreVariant 619
SAP_SynchronizeEventHistoryCriteria 620
SAP_SynchronizeInterceptionCriteria 620
SAP_SynchronizeTerminatedCcmsJobs 621
Process Server Parameters 622
Scripting Contexts and Implicit Objects 622
Use 622
Scripting Contexts 623
Using RedwoodScript in Objects 630
REL Functions 630
Built in functions 631
JCS Errors 644
JCS Errors 101XXX (alert) 644
JCS Errors 102XXX (api-impl) 644
JCS Errors 103XXX (api-int) 651
JCS Errors 104XXX (api) 651
JCS Errors 105XXX (as400) 652
JCS Errors 106XXX (audit) 652
JCS Errors 107XXX (cl-api) 652
JCS Errors 108XXX (cl-impl) 652
JCS Errors 109XXX (cluster) 653
JCS Errors 110XXX (configuration) 653
JCS Errors 111XXX (core) 653
JCS Errors 112XXX (db-init) 654
JCS Errors 113XXX (export) 655
JCS Errors 114XXX (infrastructure) 656

JCS Errors 115XXX (jobchainservice) 656
JCS Errors 116XXX (jobservice) 656
JCS Errors 117XXX (mailservice-api) 657
JCS Errors 118XXX (security) 657
JCS Errors 119XXX (servlet) 657
JCS Errors 120XXX (ui) 658
JCS Errors 121XXX (module-r2w) 658
JCS Errors 122XXX (agent-api) 658
JCS Errors 122XXX (model) 658
JCS Errors 123XXX (persistence) 660
JCS Errors 124XXX (script-java) 660
JCS Errors 125XXX (constraint) 661
JCS Errors 126XXX (iphone) 661
JCS Errors 130XXX (apns) 661
JCS Errors 131XXX (dblayer-sql) 661
Glossary 661
Use 661
A 661
B 662
C 662
E 663
F 663
G H I 663
J 664
K 665
L 665
M 665
N O 665
P Q 666
R 666
S 666
T 667
U V W X Y Z 667
Attributions 668
Apache Log4j 1.2.9 668
Apache Geronimo 1.0 668
Redwood Platform 6.0.10 668
Apache Derby 10.2.1.6 668
Sun JavaMail 1.2 669
Sun Java Runtime Environment 1.6.0.03 (Win32) 669

Chapter 1. Welcome

Use

Introduction

SAP CPS is an event-driven process automation and scheduling tool.

It provides the following facilities:

- Event-driven process automation.
- Asynchronous workload management.
- Complex process flows.
- Business driven schedules.
- Workload routing and load balancing.
- Monitoring and managing present and future workload.
- History, logging, and auditing of previously executed work.

There are two versions of SAP CPS, the Free of Charge version, which ships with SAP software and covers the functionality of transactions **SM36** and **SM37**, and the Chargeable version, which allows extended SAP scheduling as well as integration of non-SAP processes, amongst other things. The following table illustrates the main differences between the two version. A complete list of differences can be found in the Different Versions section of the documentation.

Capability	**SAP CPS (Free of Charge)**	**SAP CPS (Chargeable)**	**Description**
Manage ABAP jobs	Yes	Yes	Standard ABAP scheduling using XBP. With the Free of Charge version any number of SAP Systems can be managed but only within each instance/client combination. No cross SAP client, cross SAP System or non-SAP scheduling is possible with the Free of Charge version.
Basic SAP Solution Manager integration	Yes	Yes	SAP job monitoring. Synchronize and visualize CCMS or Solution Manager Monitors (XAL/XMW). Also allows system-wide monitoring for one user.
Basic SAP BI integration	Yes	Yes	Submit and monitor BI process chains and info packages. Visualize parent and child chains.
SAP PI/XI integration	No	Yes	Control SAP communication channels and document conversions.
Extended SAP Solution Manager integration	No	Yes	Allows business process management (BPM) and change management integration.
SAP BI Process Chain restart	No	Yes	Allows restart of BI process chains from given point within the chain.

SAP Financial Closing Cockpit integration	Allows automation of closing processes to be managed by Closing Cockpit users. Requires SAP Financial Closing Cockpit from SAP.		
Backwards compatibility with older XBP versions	No	Yes	Allows use of Redwood transports to enable enhanced capability for older versions of XBP (SAP Basis 4.6A and lower). The Free of Charge version will be limited to the capabilities of the XBP release. For more information on XBP releases, see the Understanding The Different XBP Interface Releases section.
Load balancing of SAP jobs	No	Yes	Targets workload at lowest usage SAP application servers. Does not just use *round robin* approach.
Import / Export SAP CPS objects	No	Yes	Migrate processes from DEV to QA to production.
Automate Operating System jobs	No	Yes	Schedule non-SAP processes across your entire data center. Requires relevant OS platform agent on each managed server (Windows, Linux, UNIX, and HP OpenVMS).
Process Server (Platform Agent) based load balancing	No	Yes	Uses Redwood monitoring tree to determine where to run workload (CPU, paging rate etc). Requires more than one Platform Agent.
Manage non-SAP application jobs	No	Yes	Using suitable interface for application (typically CLI).
Manage file events	No	Yes	Detect incoming files and trigger workload.
Manage non-SAP events	No	Yes	Detect and respond to events external to SAP (for example job status change).
RedwoodScript	No	Yes	Allows programming access to SAP CPS using Java.
Send email	No	Yes	Job definition to send an email directly from SAP CPS
Send SNMP trap	No	Yes	Job definition to send an SNMP trap directly from SAP CPS

See Also

- Getting Started - Using SAP CPS
- Automate your SAP System Landscape
- Automate non-SAP processes

Chapter 2. Release Notes

The current topic contains the release notes for SAP CPS build M28.49-54820 built on Tue Jan 03 14:39:49 2012.

Browser Versions

The following browser versions are supported with M28:

- Firefox 3.x
- Internet Explorer 6, 7, and 8

Other, older, or newer browsers are not supported and may or may not work.

Reverting from Chargeable back to Free of Charge Version is NOT Supported

It is not supported to revert from a Chargeable to a Free of Charge version, you *must* remove the software from the application server and database, as if you were installing on a clean system, prior to re-installing the Free of Charge version. Note that job data cannot be kept.

Run As Fields and Special Characters in Passwords

Customers have reported that when they specify a user/password in a *Run As User* field in a job definition and have special characters (*/* and @) used in the *Run As User* field in their passwords, the jobs reach status *Error*. This is due to the fact that both */* and @ are used as field separators in *Run As User* fields.

Specifying a username with password in a *Run As User* field will be deprecated in favor of credentials, which offer a secure central store for passwords.

Redwood Platform and Oracle Prerequisites

The documentation previously stated that you should use the roles `CONNECT` and `RESOURCE` to grant the Oracle user that SAP CPS uses to access the database the required privileges. After further investigation, it turns out that these roles grant not only too many privileges, but also hidden privileges. The `RESOURCE` role includes `UNLIMITED TABLESPACE` which is not what most customers would want. Note that you can only see `UNLIMITED TABLESPACE` in the user's session privileges after granting the user the `RESOURCE` role. When you grant `UNLIMITED TABLESPACE` to a user, then grant and revoke the `RESOURCE` role, the user loses the `UNLIMITED TABLESPACE` privilege!

Note that no issues have been reported regarding this, but SAP recommends you grant the following privileges only:

- `CREATE SESSION`
- `CREATE TABLE`

Also, the user needs unlimited quota on the default tablespace for that user, as that is the tablespace used by SAP CPS.

So, to edit an existing user perform the following:

```
SQL>  sqlplus system/mypassword@orcl
SQL>  alter user masalan
  2 quota unlimited on users;
SQL>  revoke CONNECT, RESOURCE from masalan;

Revoke succeeded.

SQL>  grant create session, create table to masalan;
```

To create a user with the required privileges:

```
SQL>  sqlplus system/mypassword@orcl
SQL>  create user masalan identified by mypassword
  2 default tablespace users
  3 quota unlimited on users;
SQL>  grant create session, create table to masalan;
```

Note that in these examples, `orcl` is the connect string, `masalan` is the user and `users` is the tablespace.

Note, also, that `mypassword` is not a very safe password and just used here for illustration purposes.

Tablespace Usage - Upgrading from Version 7

When you upgrade from Cronacle version 7.0.4 to SAP CPS, you need to plan for potentially greater tablespace usage.

SAP CPS version 8.0.0 has the following differences that may affect tablespace sizes:

- use of a UTF characterset is highly recommended; it potentially uses more space to save characters than others - it has the advantage, though, that it can store a much greater number of different characters
- objects can have longer names, this was increased from 30 to 80 characters
- default job definitions have more job parameters which provide new functionality

Killing RedwoodScript Jobs

When you have short infinite loops, like `{while (true){}}` you will not be able to kill the RedwoodScript job. To stop this job you will have to restart SAP CPS. Note that this problem does not occur for `{while (true){Thread.sleep(1);}}`.

FileInUseCommand Process Server Parameter Changed Default Value on Unix

The FileInUseCommand process server parameter now has the full path to the fuser file as the default value on Unices (except GNU/Linux), just like it was the case for Oracle Solaris.

The new default value has become {{/usr/sbin/fuser ${Filename}, }}it was set to {{fuser ${Filename} }}on Unix, except for Oracle Solaris and GNU/Linux.

This change was made to allow customers to remove `/usr/sbin` from the{{ ${PATH} }}of the user. The default for GNU/Linux is {{lsof ${Filename}. }} Note that `lsof` is on default user path, usually located in `/usr/bin`.

SAP Application Return Codes

SAP application return codes will now be ignored by default with new installations. When you upgrade from M31 or earlier, a registry key will be created and set to `true` so as not to change the behavior of SAP CPS. Ignore-ApplicationReturnCode Registry Key

The registry can be set to `true` or `false` at the following location:

`/configuration/sap/xbp/IgnoreApplicationReturnCode`

Dates are Lenient

Dates in SAP CPS are lenient, which is the default in Java. This means that `32/12/2011` is treated as `01/01/2012`.

SQL Server Deadlocks

To prevent deadlocks on SQL Server, SAP highly recommends enabling row versioning-based isolation. You can enable this by turning the database parameter `READ_COMMITTED_SNAPSHOT` to on.

Perform the following Transact-SQL statement to enable `READ_COMMITTED_SNAPSHOT`:

```
ALTER DATABASE <DB> SET READ_COMMITTED_SNAPSHOT ON;
```

Note

In order to set `READ_COMMITTED_SNAPSHOT ON` or `OFF`, there must be no active connections to the database except for the connection executing the `ALTER DATABASE` statement.

Job Chain Status

When the last step of a job chain is *Skipped*, *Ignored* or *Disabled*, then the job chain reaches status *Completed*. Customers have asked for the status of the job chain to reflect the status of the last executed step.

A registry key and comment keywords have been introduced to allow you to toggle the behavior. The keyword in the Comment of the job chain definition has precedence over the registry key.

The following comments can be used to enable the new behavior:

`@useLastExecutedStepStatus=true`

To disable, use the following:

`@useLastExecutedStepStatus=false`

The registry key that governs the default behavior:

`/configuration/jcs/jobchain/useLastExecutedStepStatus`

The registry key should be set to either `true` or `false`.

Preconditions

Preconditions are executed before the job is persisted, or saved in the database. This has the effect that it is not possible to access itself or, when it is a job chain, any other job in the same job chain.

Job Run Order Change

If there were multiple jobs that could run at a single moment in time, then they would run in the following order:

1. Based on priority of the job, (highest first)
2. Based on the scheduled start time of the top level parent of the job, (oldest first)
3. Based on the scheduled start time of the job, (oldest first)
4. Based on the job id, (smallest first)

The problem here is that the scheduled start time of the job is occasionally updated, and when this happens this can change effects the order that the jobs should run in. This was not the expected behavior and has been changed.

To resolve this issue, the defined order that the jobs run in will be:

1. Based on priority of the job, (highest first)
2. Based on the requested start time of the top level parent of the job, (oldest first)
3. Based on the requested start time of the job, (oldest first)
4. Based on the job id, (smallest first)

The change is essentially that we use the *Requested Start Time* instead of the *Scheduled Start Time* when comparing jobs with identical priority, as the latter can be updated during the life-cycle of the job.

OnChange Trigger

The OnChange trigger no longer fire for the transition from `Quueued` to `LockWait`.

Firefox 3.6 and Java Plugins

Due to a change made by Mozilla Corporation, the Mozilla Firefox browser 3.6 and higher requires a Java plugin of version 6 (also known as 1.6) update 10 or higher. Earlier versions of Firefox support both the old and new plugins. The Java plugin is used in SAP CPS to run Java applets like the job chain editor, for example. There is a HTML version of all Java applets available as well. If you experience issues with SAP CPS Java applets, check your Java plugin is working correctly on this page [http://java.com/en/download/help/testvm.xml] and that you have a recent version, at the least Java 5 (Java 6 update 10 or higher is required by Mozilla Firefox 3.6 and higher).

Only Sun/Oracle Java Plugins 5 (also known as 1.5) or higher are supported, excluding preview and pre-release versions. The icedTea web browser plugin, for example, is known not to work. A preview version of Java 7 did not work.

Debian-based Systems and Java Networking

There is a bug in the latest Debian-based GNU/Linux systems which prevents Java applications from listening on network ports.

This was introduced via the sysctl-setting NetBase

`net.ipv6.bindv6only=1` in the file `/etc/sysctl.d/bindv6only.conf` is the default setting in Debian squeeze.

This setting breaks networking in Java, and any traffic will always result in a `java.net.SocketException: Network is unreachable`.

Setting the value to `0` fixes this issue. More information is available in the Debian bug-tracking system [http://-bugs.debian.org/cgi-bin/bugreport.cgi?bug=560044].

Import Jobs and Clusters

In previous releases, when you ran a job to import a car file, for example, the file would be saved in the *current working directory* of the cluster node. This can cause issues if the actual job is run on another node, as it cannot access the file.

A new registry key `/configuration/jcs/upload/UploadedFilesPath` has been introduced to specify the location where uploaded files, not only car files, are saved. This directory must be accessible to all nodes in the cluster.

SQL Server JDBC Driver Settings

SAP strongly recommends you enable response buffering in the jdbc driver connection string. You also need to remove `selectMethod=cursor` if set, as this setting overrides the response buffering setting.

Open [<drive>:]/usr/sap/<sid>/<instance>/j2ee/configtool and navigate to *secure store* and append `responseBuffering=adaptive` to the *jdbc/pool/<database_name>/Url*

For example:

In the configtool, navigate to *secure store* and append responseBuffering=adaptive to jdbc/pool/MSLN/Url

In this example, the connect string becomes:

```
jdbc:sqlserver://mssql.masalan.com:1433;databasename=MSLN;responseBuffering=adaptive
```

Note

The sqlserver jdbc connection strings have semi-colon (;) separators; you must be especially careful when changing the database connection string.

SAP Interception

SAP Basis releases 7.0 SP14 to SP17 can have issues with interception when configured in XBP 3.0 mode. Please see SAP Note 1263218 for more information.

As a workaround, you can set the Interception mode to XBP 2.0 with the ABAP program `INIXBP2`. All other XBP 3.0 features will still be available.

JVM Time Zone Settings

The central server uses the time zone configuration for the underlying JVM. You must make sure that the time zone settings of the JVM are up-to-date. The easiest way to make sure this is the case it to use the latest version of the JVM in question. This is especially true for Daylight Saving Times, which have been changed in various countries in the last years.

HTTP Worker Threads

You need to allocate `2n` HTTP worker threads where `n` is the number of concurrent users that you want to support. This is because the SAP CPS user interface uses 2 HTTP connections per user. This amount does not take into account threads used by other applications. SAP recommends a base of 20 + `2n` with a minimum of 50.

An SAP Note for a similar problem exists, see SAP Note 1169442. A lack of HTTP Worker threads will manifest itself as very poor performance, with response times in the (keep alive timeout) + 1-2 second time range. The default keep alive timeout is 30 seconds.

Oracle Globalization Support

Oracle NLS settings allow you to support different locales for various database activities. Sorting data, data collation, can be controlled with `NLS_SORT` and `NLS_COMP`.

SAP CPS expects one of these combinations:

- `NLS_SORT` set to a linguistic sort (French, for example) and `NLS_COMP` to `LINGUISTIC`
- `NLS_SORT` and `NLS_COMP` set to `BINARY` (or variant like `BINARY_AI`)

Other combinations are not supported, as they can cause unexpected behavior.

Clients Require Time Zone Data

Time zone information is client-dependent and has to be maintained in each client separately. The report *TZCUSTIM* imports time zone data into a client. Please refer to SAP Note 198411 for more information.

File Event Move Directories

The substitution parameters ${DateTime} and ${TimeStamp} have been deprecated and should not be used anymore. The equivalents are ${FileDateTime} and ${FileTimeStamp}, respectively.

Default Pagesize on DB2 is insufficient

The default pagesize on DB2 is too small, please set it to at least `32768`. This can be done by specifying `PAGESIZE 32768` at database-creation time.

Queues, Time Windows and Time Zones

To make sure the correct time is used when calculation opening times of queues, you should always set a time zone on a queue with a time window.

Platform Agent Network Timeout

A network timeout has been introduced to solve networking issues, when no message has been received in at most twice the MessageInterval, the platform agent will attempt to reconnect.

Fail-over Timeouts

It is now possible to configure fail-over timeouts with the SAP CPS registry key: `/configuration/jcs/clustering/lockTimeoutThreshold` the value is in seconds.

File:// links and Mozilla Firefox

The security features in Mozilla Firefox prevent it from opening `File://` links directly. This can be overruled by specifying the following in your user.js file:

```
user_pref("capability.policy.policynames", "localfilelinks");
user_pref("capability.policy.localfilelinks.sites", "http://pr1.masalan.com:50200" "http://pr2.masalan.com:50
user_pref("capability.policy.localfilelinks.checkloaduri.enabled","allAccess");
```

In the above example, Masalan Inc. has three instances of SAP CPS it would like to enable `file://` links for:

`http://pr1.masalan.com:50200, http://pr2.masalan.com:50200, and http://dv1.masalan.com:50200`

The user.js file is usually located in the following directories:

On Windows

```
%AppData%\Mozilla\Firefox\Profiles\xxxxxxxx.default\
```

On Linux/Unix

```
~/.mozilla/firefox/xxxxxxxx.default/.
```

On MacOS X

```
 ~/Library/ApplicationSupport/Firefox/Profiles/xxxxxxxx.default/.
```

Note

`xxxxxxxx` is 8 random characters.

Fail-over Timeouts

It is now possible to configure fail-over timeouts with the registry key: `/configuration/jcs/clustering/lockTimeoutThreshold`

Time Windows and the Custom Library

To use time windows with period functions, including built-in time windows, the `Custom` library needs to be visible.

Shell

To use the shell, you need the privilege *System_Shell*, this was previously not the case.

Transport Files

New transport files are shipping with this release, these are required for controlling Mass Activities and can be installed alongside older versions.

Changing the SAP System of a Job with Default Queue

If the queue is already set, for example as the default queue, you want to change the SAP system where the job runs, and the process server of the target SAP system does not serve the default queue, you must first clear the *Queue* scheduling parameter. This limitation has been introduced to avoid typos and to allow you to choose SAP systems for which process servers serve the queue.

SAP Notes

SAP Note Number	**Title**	**Description**
1161405	Accumulative Note for SAP CPS for SAP NetWeaver 7.0	Provides links to multiple SAP Notes relating to SAP CPS installed on SAP NetWeaver.

Upgrade Issues

Upgrading from 26 to 28

When you upgrade from M26 to M28, note that the default system ID in M28 contains the port number as well. When you have auto-upgrade activated, you may encounter the issue that the process servers no longer can connect to the Central Scheduler. This seems to occur when you have a locked the platform agent to the central instance. This information is stored in the *secret* file, located in `${install_dir}/net/instance/default/private/secret` on the platform agent host. You have to regenerate the file again when this problem occurs.

SAP Job Definitions

When you are upgrading from a previous version, please note that a number of optional parameters and constraints in SAP job definitions are added to the built-in SAP job definitions only, not to your SAP job definitions. If you want to make use of these, for SAP load balancing for example, you will have to add these parameters yourself or import the SAP jobs again; constraints cannot be added manually and require you to import the job definitions again. This has been done to significantly increase the upgrade speed.

The following parameters are affected:

- `TARGET_GROUP` - the target group of SAP application servers, used for load balancing
- `TEMPORARY_VARIANT` - allows you to use a variant that is deleted after the job has run (XBP 3.0 is required)
- `CLIENT` - has been added for Mass Activities
- `JOBCOUNT` - out parameter for use in job chains or Solution Manager

SAP CPS does not require these parameters if you do not want to make use of the functionality.

SAP Job Names

In **SM36** the job name is always uppercased, however, there are many other ways in which jobs are created with mixed-case names. Because of this, the job definitions *SAP_ImportCcmsJobs* and *SAP_GenerateCcmsJobDefinition* do not uppercase the job name automatically anymore.

Events and Job Files

When you upgrade SAP CPS and your existing version is M22 or earlier, make sure that every process server has a *DataRootDirectory* parameter set. If this parameter is not set the new default value will be different and point to a version independent value. This can influence file events, as they will loose their state causing potential re-raising of file events. This issue does not play a role if you upgrade from M23 or later, as your default directory is already version independent.

Application Renames

During the upgrade, if applications called *System* or *Redwood* are found, they will be renamed. This is due to the fact that these are now reserved words.

When this occurs, the following actions are taken:

- The offending application is renamed to *<name><number>*, where name is the original name of the application, and number is a random number.
- An operator message will be created for each renamed application, giving the old and new names of that application.

Platform Agents on Linux

On Linux, Platform Agents require glibc 2.4 or higher.

Platform Agents on Tru64

A UTF-8 character set is required on Tru64 systems, like for example en_US.UTF8, or the platform agent will fail to start with the following error:

```
<install_path>/<version>/bin/network-processor: OS initialization failed
error  2011-05-18 11:50:52 CEST [<pid>-platform-agent] Scheduler network-processor instance '<instance>' died
```

- <install_path> - installation directory
- <version> - version directory; for M28.32 this would be 0_28_32, for example
- <pid> - process id of the platform-agent process
- <instance> - name of the instance

JCo Dependency Issues

On UNIX and GNU/Linux

When you install SAP CPS on GNU/Linux, note that the SAP JCo libraries depend on libstdc++. Use ldd to make sure the dependencies are met.

On Mac OS X, you use otool, other UNIX systems have their own tools, please refer to the documentation.

Windows

Please refer to SAP Note 684106 for information on resolving JCo dependency problems on Windows.

Chapter 3. New Features

This topic lists the new features for SAP CPS build M28.49-54820, built on Tue Jan 03 14:39:49 2012.

Import Jobs and Clusters

In previous releases, when you ran a job to import a car file, for example, the file would be saved in the *current working directory* of the cluster node. This can cause issues if the actual job is run on another node, as it cannot access the file.

A new registry key `/configuration/jcs/upload/UploadedFilesPath` has been introduced to specify the location where uploaded files, not only car files, are saved. This directory must be accessible to all nodes in the cluster.

Audited Objects have and Audit Trail tab

When an audited object is modified, the next time you open the object editor, the object will have an *Audit Trail* tab, which allows you to undo changes, should you require to do so.

Improved Security with sudo

When you use sudo for user-switching on Unix, you can now specify your chown command and check parameters for validity, making sure a rogue user who gained access to the SAP CPS OS account cannot access files owned by `root`. SAP CPS ship with an example `chown.sh` script which can be used as a basis. See Configuring Platform Agents for more information.

Platform Agent Network Timeout

A network timeout has been introduced to solve networking issues, when no message has been received in at most twice the MessageInterval, the platform agent will attempt to reconnect.

Fail-over Timeouts

It is now possible to configure fail-over timeouts with the SAP CPS registry key: `/configuration/jcs/clustering/lockTimeoutThreshold` the value is in seconds.

Permalink Module

The Permalink Module allows third-party applications to interact with SAP CPS.

Links in Comment Fields

The *Comment* fields of all objects now support the following URL's:

- http://
- https://
- file://

Note

Due to security settings in Mozilla Firefox, `file://` URL's must first be enabled, see the Release Notes for more information.

SAP SNC Support

SAP CPS can now use Single Sign On (SSO) over RFC connections, making the connection encrypted. See the Connect to an SAP System with SNC topic for more information.

Advanced Auditing Rules

You may now define advanced auditing rules, for objects of a given name or being in a specific application.

Advanced Search for all Objects

SAP CPS now offers the quick and easy search feature available in previous versions for job search on most objects. When you are in an object view, for example in *Definitions* > *Job Definitions* you can quickly locate the job definition by entering a distinctive part of its name into the search field. The search field is context-sensitive, so it will only search for objects in the currently selected tab.

Easy Retrieval of Process Server Logs

Process servers and jobs now offer a new action to retrieve log files and system information, this is especially useful for support queries.

Controlling Communication Channels with SAP PI

SAP CPS now allows you to control communication channels via the SAP PI interface. This functionality gives you advanced control of when which transformation take place. This functionality requires a Chargeable license.

New Features in M26

These features appeared in M26 and are listed here for your convenience.

SAP Applications API Integration

SAP CPS has an RFC server that you can use to connect to SAP CPS from ABAP stacks via an RFC destination and submit jobs from Closing Cockpit or Solution Manager. Other SAP components are being adapted to make use of this new feature.

Mail Job Definition Type

The Mail job definition type has been introduced to allow you to create custom emails from within a job chain.

Period Functions

It is now possible to create period functions. You have to specify a library containing a valid JAVA 1.4.2 package that can be used with period functions. Period functions allow you to create complex schedules.

Reports

The reporting has been improved in this release. The Report object allows you to specify filters or custom queries and displays the output in csv or html.

Credentials

Credentials have been introduced to allow you to securely save login information for external systems. There is also a new *virtual* user, which allows you to work with many different user names and passwords across several systems by mapping them to one virtual user. Jobs will then execute as the correct user on any of the systems, regardless of the real user name on the target system. You could have a virtual user `db_user` that would be `jdoe` on server1 and `jdupont` on server2.

Dashboard

The dashboard has been introduced to allow you to monitor components and systems from within one window. The dashboard allows you to have graphs and tachometers to display the current status.

Library and Triggers

The scripting module has been extended with libraries and triggers. Libraries provide a central location for your RedwoodScript packages and are available in all objects that provide scripting capabilities. Triggers allow you to execute code globally when a job reaches a specific status or an object is changed by a user, for example.

Isolation groups

Isolation groups allow you to isolate objects from users. They are also used to isolate SAP Systems from each other when you have the Free of Charge license.

Chapter 4. SAP CPS

Use

SAP CPS provides landscape-wide, real-time, event- and time-driven job scheduling in Application Server (AS) ABAP of SAP NetWeaver AS. It enables process automation by managing all background processes that involve one or more applications across the whole system landscape. SAP CPS is an event-driven process automation and scheduling tool with asynchronous workload management, routing, and load balancing of complex process flows.

There are two general licenses available, the Free of Charge license allows scheduling of SAP-only processes, the Chargeable license includes non-SAP processes.

Additional features may require a specific license item, you can check your license by navigating to *Configuration > License Keys*, some features require specific license items with specific values.

SAP CPS provides extended SAP as well as non-SAP support. When a feature is not available, it usually means that you do not have the required license item. If you are unsure of the features that have been licensed to you, please consult your SAP CPS Administrator, Customer Support, or the Sales Representative.

Processes follow business-driven schedules and you monitor them and manage present and future workload from one central location. SAP CPS provides process history, logging, and auditing.

SAP CPS is a JEE web application consisting of:

- A central system that runs the user interface and server software.
- Process servers that manage work in various systems
  - via External Background Processing (XBP) for SAP Systems.
- A database that stores static data, definitions, and operational data.

These all run as part of the Scheduler web-application, the Platform Module offers platform process servers that run natively on remote systems and allow you to integrate non-SAP processes into the process-scheduling system.

Implementation Considerations

SAP CPS is suitable for automating the execution of simple or complex business processes that comprise multiple steps and involve various ABAP applications across the system landscape.

As a process automation solution, it allows you to:

- Speed up the execution of automated processes by minimizing the stand-still time during business process changes. It provides an automated transition between the steps of a business process to ensure a smooth and reliable process flow.
- Extend event-driven process automation.
- Implement business-driven schedules
- Maximize the degree of utilization of system resources
- Automate synchronous as well as asynchronous workload management
- Maximize workload routing and load balancing

- Monitor and manage present and future workload
- Reduce the number of human errors resulting from manually completed tasks
- Simple as well as complex process flows are executed faster
- Extend history, logging, and auditing of previously executed work.

Integration

By using a set of external SAP interfaces, SAP CPS integrates with remote SAP Systems and the Central Monitoring system. For more information, see Technical System Landscape.

Features

SAP CPS provides the following features:

- Central user interface (UI)
- The central point of control over the central process scheduling infrastructure is designed for both administrators and business users. It allows:
    - Business users to schedule, submit and monitor all jobs in the landscape.
    - Administrators to centrally manage SAP CPS and the whole central process scheduling landscape.
    - Event- and time-driven scheduling and execution of jobs and complex job chains running across several systems in the job scheduling landscape
    - Job definitions and jobs, which allows you to re-use part or all of the job
    - Mechanisms for scalability and advanced load balancing

Note

In contrast, in the SAP conventional job scheduling infrastructure, you can schedule, execute and monitor jobs only on the local system (transactions **SM36** and **SM37**). For an introduction to the Scheduler UI, see Getting Started.

The abstract logic of the work to be performed is stored in a job definition. When a job definition is scheduled with specific parameters and start conditions, a concrete instance of this job definition, called job, is executed. The same job definition (or work done) can be scheduled with various parameters and start conditions. For the execution of complex business tasks, you can define job chains. For more information, see the Automating Processes Across the System Landscape section.

Note

In contrast, the SAP conventional job scheduling infrastructure does not distinguish between these two types of job information.

Activities

Task	More Information

Configure a remote SAP System	To set up remote SAP Systems see: Configuring an SAP System for Central Process Scheduling
Connect SAP CPS to remote SAP Systems	To connect to a configured SAP System see: Connecting to SAP Systems
Submit a Job or Job Chain	To submit jobs see: Submitting a Job or Job Chain
Import your Jobs	To import jobs from an SAP System see: Importing CCMS Jobs
Create a sequential job chain with your imported SAP jobs	To create a sequential job chain see: Creating a Job Chain with Sequential Jobs

More Information

Architecture

Getting started

Administering the Scheduling Landscape

Automating Processes Across the System Landscape

Monitoring Automated Processes

Scheduling Processes Outside of SAP

Reference

Chapter 5. Architecture

This section provides information about the architecture of SAP CPS:

- Technical System Landscape An overview of the integration of SAP CPS in the whole scheduling landscape
- Central Scheduling Concepts An introduction to the basic concepts and components of the Central Scheduling system where SAP CPS runs.
- High Availability examines high availability for SAP CPS.

Technical System Landscape

Use

SAP CPS runs in AS Java of SAP NetWeaver AS. The system where it runs is called the SAP CPS server system. It connects to the remote systems where jobs should be executed.

SAP CPS provides the Scheduler user interface running on the SAP CPS server system. It is the central point of control for the all the jobs in the job scheduling landscape. By using the Scheduler UI, you schedule jobs on the SAP CPS server system. When the start conditions for a job or job chain are fulfilled, SAP CPS executes it in the respective remote system.

Integration

By using a set of external SAP interfaces, SAP CPS integrates with:

The remote SAP systems where jobs should run and be monitored

The SAP system for Central Monitoring where it reports monitoring data.

Integration for Job Execution and Monitoring

SAP CPS integrates with remote AS ABAP systems, including SAP Business Information Warehouse (BW) systems, by using the following external interfaces:

- BC-XBP (External Background Processing) Allows SAP CPS to execute and monitor jobs and job chains in the remote AS ABAP systems. XBP provides SAP CPS with all the capabilities of the conventional AS ABAP scheduling infrastructure, such as job scheduling and starting, job interception and monitoring, and so on.
- BW-SCH (Business Information Warehouse - Job Scheduling) Allows SAP CPS to schedule BW InfoPackages and process chains in the BW system.

You have to explicitly connect the SAP CPS server system to the remote systems when you want to execute and monitor jobs, InfoPackages and process chains.

SAP CPS uses Remote Function Calls (RFC) to connect to the remote systems.

All data that is necessary for triggering activities in the system landscape, such as job definitions, event information, and so on, is persisted in the job repository in the SAP NetWeaver database.

Two-Way Integration for Monitoring Data

SAP CPS can read monitoring data from the remote systems where jobs are executed and report it to the SAP system for Central Monitoring. This two-way integration is enabled by the following external interfaces:

- BC-XAL (External Interface for Alert Management) Allows SAP CPS to read or set the properties and values of monitoring attributes, as well as read or confirm alerts in the remote systems.
- BC-XMW (External Monitoring Write) Allows SAP CPS to transfer monitoring data to the Computing Center Management System (CCMS).

Integration for Job Monitoring

For more information about the components of SAP CPS, see the Administration Guide, Getting Started.

See Also

Configuring the Central and Satellite Systems

Central Scheduling Concepts

Use

SAP CPS is an application that runs on Java Application Servers like the SAP NetWeaver Application Server (AS) Java and consists of:

- A central system that runs the user interface and server software.
- A database that stores static data, definitions, and operational data.

Note

These all run as part of the Scheduler web-application.

The following topics briefly discuss basic SAP CPS concepts:

1. Jobs, Job Chains, Job History
2. Queues and Process Servers
3. Distribute Workload
4. SAP Systems
5. Events and Event Definitions
6. Scheduling
7. Applications

Jobs, Job Chains, Job History

Use

Jobs lie at the core of SAP CPS. Jobs can be submitted by users, as the result of a particular event (for instance the arrival of a file, or a trigger in a remote system), or based on a business schedule. A job can be as short and simple as a line of ABAP code that runs on a single system, or as complex as a multi-day mass activity that runs across dozens of machines.

Jobs and Job Definitions

SAP CPS not only allows you to submit simple jobs, it allows you to create job chains with complex parameters. Custom job definitions with complex parameters can be written that are re-used for different tasks. Consider file transfer as an example; when there are only one or two files to be transferred, then writing a job for each file transfer that contains all the commands to transfer the file (including the file name) might be sufficient. When there are dozens (or even hundreds of files), a better approach is needed.

In SAP CPS, this could be handled by creating a job definition called *Transfer File* that has four parameters: *System to transfer from*, *Directory to transfer from*, *System to transfer to*, *Directory to transfer to*. This would allow operators to easily add new file transfers to the schedule, and clearly see the files being transferred (and the systems involved) in the job monitor. If some improvement was made to the transfer process, for example, using secure FTP instead of FTP, then this change can easily be made by changing the job definition. If a mistake is made, this can be fixed by changing the parameters of the job (through the job monitor), rather than editing the job itself.

Note

FTP support requires the Platform Module, which allows you to schedule programs on remote UNIX and Windows servers.

Submitting a job definition allows you to specify values for the parameters, creating a job. Each job can then be monitored throughout its life cycle: from being submitted, to running, to finishing (either in a *COMPLETED* or *ERROR* state) to finally being deleted. The job definition approach means that you can have as many jobs running as you need, rather than having to create new jobs as business needs change.

Job chains are job definitions of a specific type. They are made of one or more calls to job definitions, named jobs, and are comparable to multi-step SAP jobs. Complex job dependencies can be met with job chains, steps and jobs can be restarted when a job fails, for example.

Job History

Jobs are saved in the system according to the keep clause. The keep clause is set on a job definition or job chain basis. The default keep clause is to keep jobs forever. The keep clause can be expressed in job versions or dates. For example, you would like to keep a jobs from a specific job definition for 12 months, each time the job completes, the system will check if there are old jobs that need to be deleted and delete them for you.

You can also choose to manually handle job history, the context menus of job definitions and job chains have an action *Delete final state jobs*. The output of deleted jobs is regularly deleted by the system job definition *System_DeleteJobFiles*.

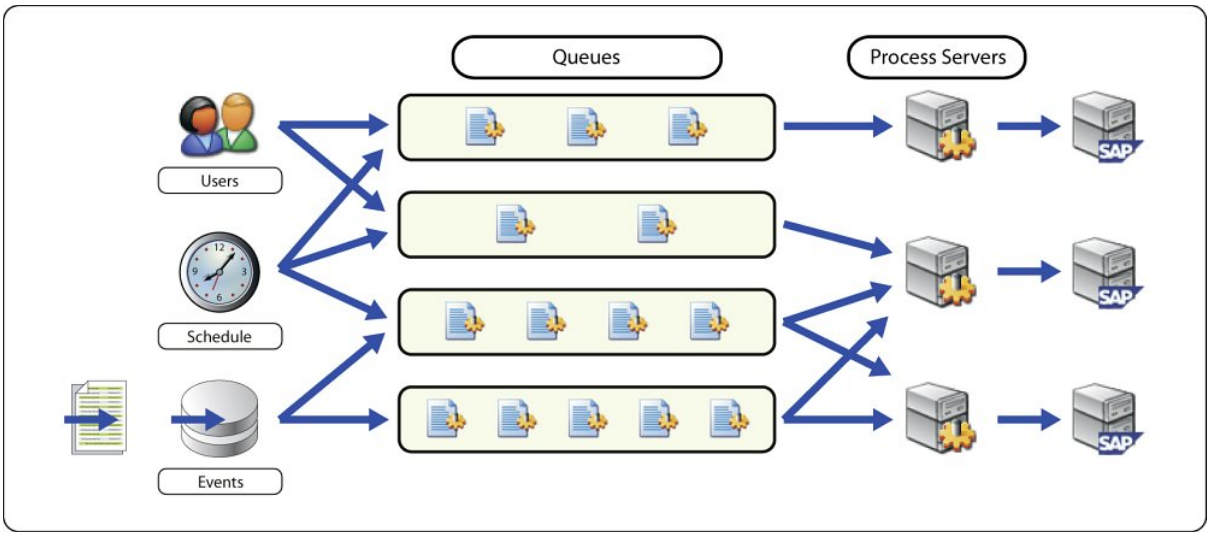
See Also

- Submitting a Job or Job Chain
- Monitoring and Managing Jobs

Queues and Process Servers

Use

The following figure shows the main architecture behind SAP CPS.



Queues

Groups of jobs are managed using queues. Queues distribute jobs to process servers, where the jobs actually run.

You can use queues to limit the number of jobs that can run simultaneously, distribute workload across multiple systems, and provide fail-over facilities. One queue can be attached to multiple process servers, in which case SAP CPS decides which process server to run the job and when the job starts. You can attach a process server to multiple queues, in which case it can run jobs from any of those queues.

Queues have limits attached to them. These limits can be dynamic or static. Dynamic limits are set based on performance measurements made by the process servers. Static limits are based on the number of jobs running in the queue. If a limit on the queue is exceeded, then the queue is held. This prevents new work from starting, but existing work is left to complete. These limits are re-evaluated as new performance measurements are made, or existing running jobs complete.

You need privileges to view process servers and queues. A queue might appear held to you while it is open to others; this is usually due to the fact that you cannot see any of the running process servers that are serving the queue.

Note

SAP recommends you to specify a time zone on a queue, especially when you tie a time window to it to make sure the correct time is used.

Process Servers

Process servers are responsible for running and monitoring jobs. They run jobs in local and remote systems. Process servers host services, which allow jobs of different types to be run and are used to connect to remote systems, like an SAP System, a Windows or Unix system.

Note

Connecting to Windows and Unix systems requires the *ProcessServerService.OS.limit* license key to be set to a number, you also need a free slot in *ProcessServerService.External.limit*.

The different types of jobs supported by SAP CPS include:

- SAP ABAP jobs (via the XBP interface) in R/3 and InfoPackages and Process Chains via Business Information Warehouse - Job Scheduling (BW-SCH) in BI.
- System jobs (locally) for maintenance tasks.
- SAP Java jobs via Java External Background Processing (JXBP) in the SAP NetWeaver AS Java.
- CMD, KSH and many others for platform jobs

There is one process server for each SAP System or Platform Agent in the landscape. This allows Platform systems, SAP Systems and interfaces to be managed in a granular manner. For SAP Systems, this also makes it possible to manage scheduling on both a per-system and a per-interface basis.

See Also

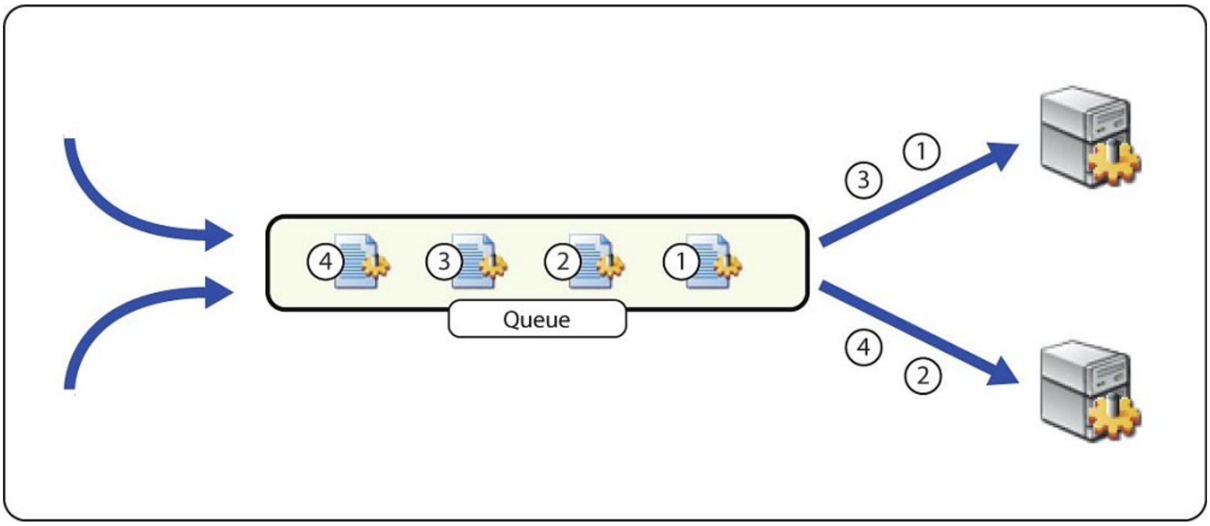
- Using Queues
- Using Process Servers

Distribute Workload

Use

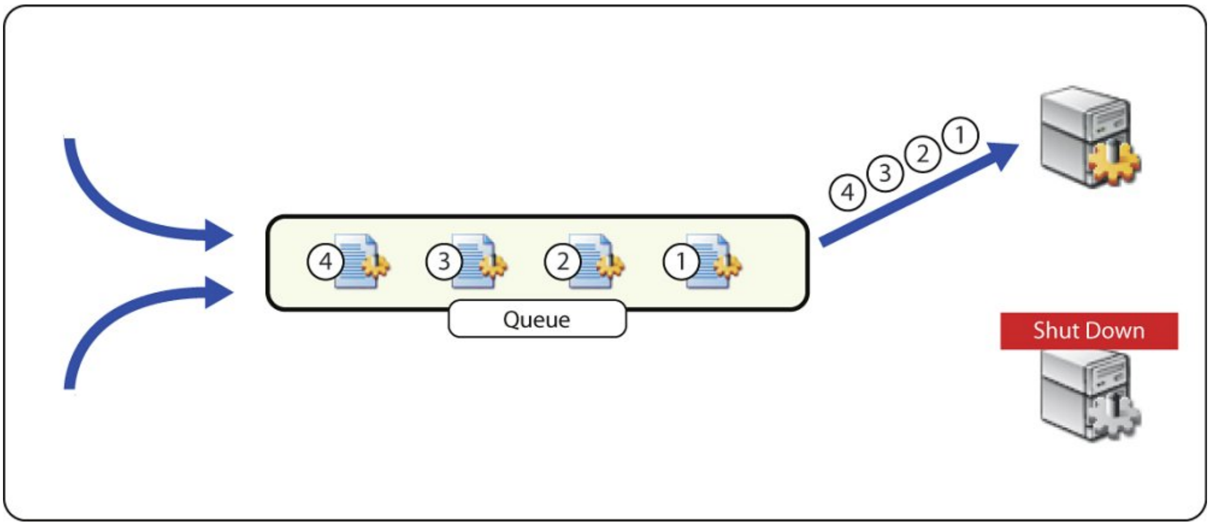
Queues distribute work across a larger system. If one or more machines can perform a task, then queues can be used to distribute work among these machines by attaching a process server to each machine, and a queue to these process servers. Jobs are automatically distributed to all the process servers as they start.

The following figure shows a queue and two process servers working normally.



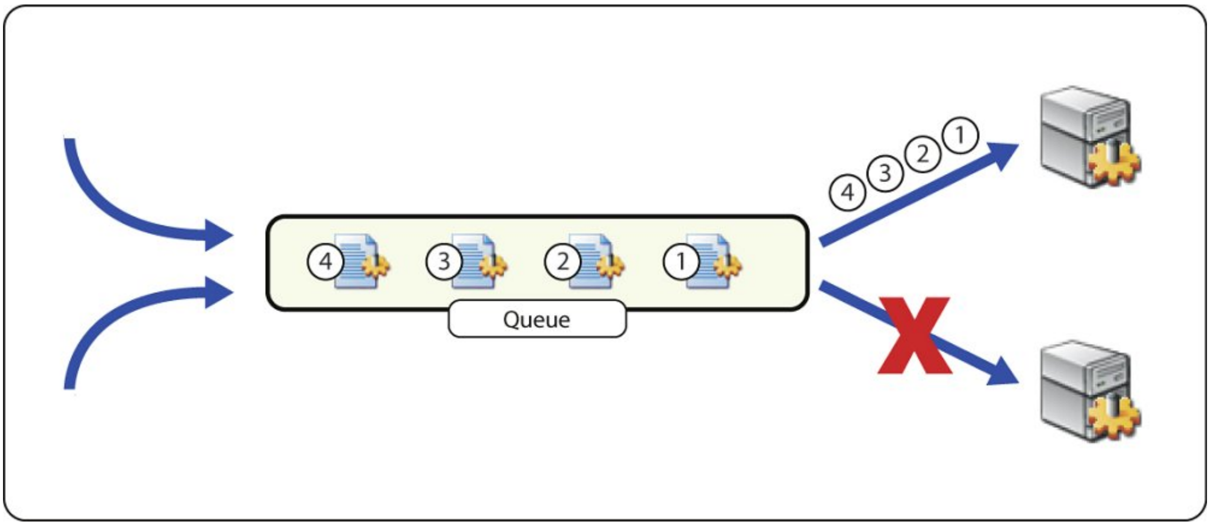
This scenario also provides for fail over in both planned and unplanned scenarios.

In a planned scenario (for example, an upgrade of the operating system is required), the process server is first shut down gracefully: all running jobs are allowed to complete, but no new jobs start. Jobs now automatically start on one of the other process servers until the planned work is completed and the process server restarted. The following figure illustrates a planned down-time scenario.



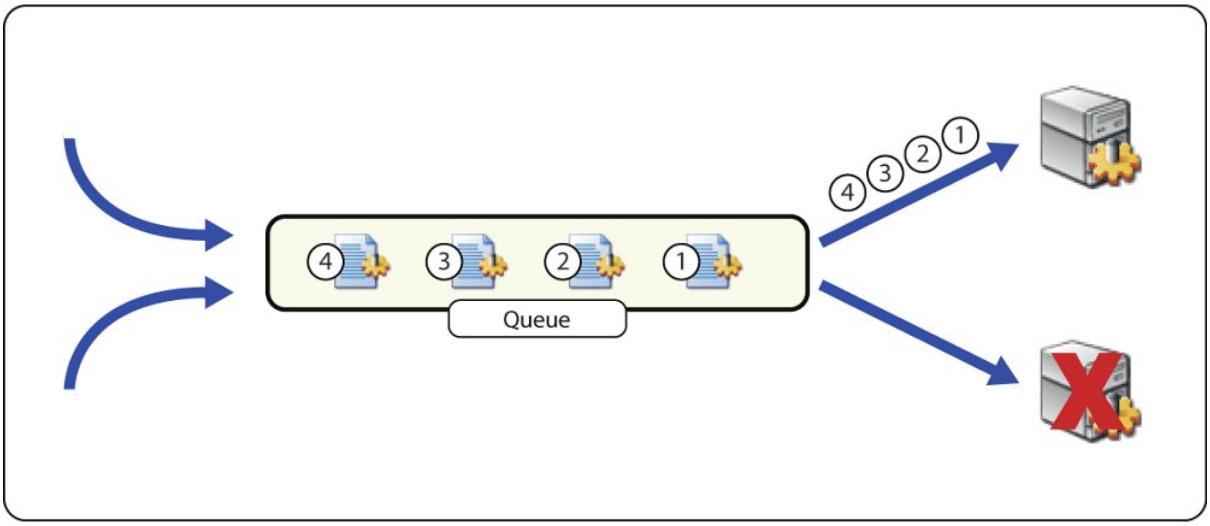
In an unplanned scenario, the process server is unreachable. This can be due to network failure, or a hardware failure.

The following figure illustrates an unplanned, network-related down-time scenario.



In the case of a network failure, jobs continue to execute, and complete on the machine. The process server attempts to notify the central system that the job has completed. It continues to retry this operation if it fails.

The following figure illustrates an unplanned, system-related down-time scenario.



In the case of a machine failure, jobs on the machine are set to the *UNKNOWN* status to indicate that their results may not be reliable (since the machine failed).

See Also

- Using Queues
- Using Process Servers
- Using Resources to Distribute the Load
- Balancing the Load on SAP Systems
- Balancing the Load on Platform Agents

SAP Systems

Use

SAP CPS communicates with the satellite SAP Systems over a number of internal and external interfaces provided by SAP.

These can be configured from a single point: *Environment > SAP Systems*.

This allows each of the following interfaces to be configured:

Interface	**Description**
XBP	Client configuration for XBP.
XBP	Job intercept rules for individual clients.
XBP	Event Control Rules.
External Interface for Alert Management (XAL)	Monitor synchronization from the SAP CCMS monitoring infrastructure.
External Monitoring Write (XMW)	Monitor synchronization to the SAP CCMS monitoring infrastructure.
Java eXternal Batch Processing (JXBP)	Control and monitor jobs deployed on AS Java.

Generic Request and Message Generator (GRMG)	Monitor the availability of technical components and entire business processes.
SAP API	Interface that allows SAP Systems to interact with SAP CPS, this is the interface that enables Closing Cockpit.

Note

SAP CPS supports the SAP Generic Request and Message Generator (GRMG) protocol, however this must be configured through Computing Center Management System (CCMS) on the Central Monitoring System (CEN) in the landscape.

The standard job definition *SAP_AbapRun* allows you to run ABAP reports in an SAP System. When you submit this job definition, parameters like ABAP report name, a variant to run the report with, SAP print and archive parameters among others are required. Once you set the parameters, this job can be submitted to an SAP System, where a CCMS job (**SM36**) with the provided characteristics is created and started. This job is then monitored by SAP CPS until its completion. If requested, the SAP job log and the spool list of the report are retrieved, and the job is deleted in the SAP System.

It is also possible to import jobs from an SAP System via the standard job definition *SAP_ImportCcmsJobs*. For each of the imported jobs, a new job definition is created, and the characteristics of the SAP job are stored as default values in the job definition. For SAP jobs with more then one step, a job definition for each step is created, and a job chain is generated that runs the job definitions. This allows you to run the SAP job in different SAP Systems, as long as they provide the same environment like variants, SAP printers, and so on.

Platform Systems

Platform systems is a generic term used to specify Unix, Windows and OpenVMS systems. SAP CPS can connect and schedule workload on these systems, provided you have the Platform Module and a spare unit in the values of the *ProcessServerService.External.limit* and *ProcessServerService.OS.limit* license keys. To be able to connect to platform systems, you also need to install platform-specific software, known as a platform agent, on the nodes. The software is available for download in SAP CPS in *Configuration* > *Software*. Platform systems can also be used for load balancing SAP Systems, this requires you to install the platform agent on the nodes where SAP is installed and requires the above module and license keys.

Note

When you install platform-specific software on the host of an SAP System, you will need additional *ProcessServerService.External.limit* and *ProcessServerService.OS.limit* license keys. In fact, you will use two *ProcessServerService.External.limit* units, one for the SAP process server and one for the platform agent process server.

Platform systems allow you to create complex process scheduling scenarios integrating SAP and non-SAP jobs in one process; reducing the time between jobs to seconds regardless of where the jobs run. They also allow you real-time load balancing across SAP servers; the load is assigned to the system that has the most free resources. The platform system load balancing is *not* round-robin-based like SAP load balancing.

See Also

- Connecting to SAP Systems
- Importing Objects from SAP Systems
- Handling ABAP Programs
- Handling Variants

Events and Event Definitions

Use

SAP CPS is an event-driven process automation tool. This means that it is possible to start jobs (or job chains) using events, rather than just having them happen at a particular time. This event-driven nature makes it possible to create efficient processes that do not rely on time at all. When certain conditions are met, a SAP CPS event is raised.

An event is raised, for example, when any of the following happens:

- A job or job chain completes.
- A job chain step or job completes.
- A monitor exceeds a threshold, or drops below a threshold.
- A job takes too long to complete.
- A file raises the event.
- You raise an event manually.

Jobs (and job chains) can wait for one or more events to occur, and may even be automatically submitted if an event occurs. Jobs (and job chains) can also raise events when they complete, allowing event-driven flows to be created.

Events (like jobs) have a definition separate from the event itself. An event definition can have zero or more actual events associate with it. The definition stores information like the name and the trigger for the individual events. Job definitions, monitors, and other objects are connected to the definitions of events that they depend upon, and the definitions of the events that they raise.

Each time an event occurs, a new event is created (or raised). Each event can then be individually monitored and managed, and remains visible as historical data even after all tasks related to the event have completed.

See Also

- Creating Event-Driven Schedules with Events
- Monitoring Events

Scheduling

Use

SAP CPS includes a rule-based scheduling engine that allows sophisticated schedules to be easily built and maintained. There are two primary mechanisms for scheduling in SAP CPS:

- Time windows & submit frames - Unlimited, rule-based scheduling.
- Recurrences - Time limited scheduling for irregular schedules.

Jobs submitted using either mechanism are visible in the job monitor.

Time Windows & Submit Frames

Time windows and submit frames are a rule-based scheduling method. This means that the times that a job runs at are defined by a set of rules. For example, a job can run at any of the following schedules:

- Once an hour between 8:00 am and 5:00pm (European Standard Time).
- On weekdays.
- Unless that day is a holiday in Germany or The Netherlands.
- Once a month, on the last working day of the month.

Note

For any time-based object to be precise, you must specify a time zone.

A variety of rules are delivered by default, these include:

- Definitions for holidays that do not occur on the same date each year, like Easter.
- Holiday calendars for many European countries (and their states or provinces), the United States, and Canada.
- Rules for United States fiscal quarters (445 and 544 calendars).

These rules are implemented using two complementary concepts:

- Time windows - To specify when jobs may and may not run.
- Submit frames - To specify how often jobs run.

Time windows and submit frames may be used individually, or together.

When a job is submitted using submit frames and/or time windows, the following jobs appear in the job monitor:

- All previous runs of the job (until they are deleted).
- If the job is currently running - the running job.
- If the job is not currently running - a job scheduled to run the next time the job starts.

Time Windows

A time window specifies when jobs may and may not run. A time window is *open* at a particular time if jobs are allowed to run, otherwise it is closed. Time windows are made up of *time window elements* that specify restrictions on opening times. A time window is open if at least one of its elements is open, otherwise it is closed.

A time window that is open during business hours so not at lunch-time, for example, could look like the following:

- Open between 08:00 and 12:00.
- Open between 13:00 and 17:00.

This time window is open at the following times:

- 08:00, 12:00, 13:00, 14:45 and 17:00.

Note that time window elements include their opening time (08:00 and 17:00 in this case), but do not include their closing time (12:00 and 17:00 in this case). This is to ensure that inclusion and exclusion work correctly.

Time windows can refer to one another, using inclusion and exclusion. So, another way to define the time window above would be:

Time window *Lunch:*

- Open between 12:00 and 13:00.

Time window *Business excluding Lunch*

- Open between 08:00 and 17:00.
- Closed during *Lunch*.

This method has the advantage that it is much easier to change the hours associated with *Lunch*, and the *Lunch* time window can be shared with many other time windows. It also makes it possible to use the predefined rules that come with the product:

For example, a time window that is only open when business units in Germany, the US, and Canada are open is defined below:

Time window *All Business Units Open*:

- Open from Monday to Friday.
- Closed during *System_DE_PublicHolidays*.
- Closed during *System_US_PublicHolidays*.
- Closed during *System_CA_PublicHolidays*.

Submit Frames

Submit frames specify how often a job runs, for example, once an hour, twice a month, or on the last workday of every month.

Like time windows, submit frames are made up of elements. Submit frame elements may refer to time windows, allowing different frequencies to be specified, depending on the time.

For example, a job that deletes job output files might need to run more often during peak periods:

- Every 5 minutes, during *PEAK* time window.
- Every 15 minutes, otherwise.

Note

For any time-based object to be precise, you must specify a time zone.

Recurrences

Recurrences are another method of controlling when jobs run, based on concepts similar to those you might use to schedule a business appointment. You use recurrences to schedule jobs to run on specific schedules, such as daily, weekly, monthly, yearly, or by specifying specific dates.

Time Zones

Many large organizations have systems that are distributed geographically, often in different time zones. Operators are frequently not in the same location as the jobs being managed. When working in such a situation, it is important to be able to work with jobs in the time zone that they are likely to run in.

Consider a job that runs at 5:00 pm in New York, and an operator based in London talking to a user in New York. The operator needs to be able to see that the job's time zone is *New York* and that the job runs at 5:00 pm, even though it may be 10:00 pm local time for the operator. If that operator needs to move the job to run at a new time (to allow some missing data to be loaded into a database, for example), then doing it in the appropriate (job specific) time zone is far less error prone.

The time zone of a job is the time zone specified in the submit wizard. When the job was submitted from Redwood-Script, without specific time zone, the central server time zone is used for the job time zone.

See Also

- Using Recurrences
- Creating Submit Frames
- Using Submit Frames in Jobs
- Defining Execution Times with Time Windows
- Defining Advanced Execution Times with Period Functions
- Time Zones

Applications

Use

Applications are a way of grouping the different objects in the system into logical groups, according to the purpose that they serve. For example, the process servers, queues, job definitions and job chains work with the decision support system might be put into the *Decision Support* application.

Applications are particularly useful when the scheduler is being used to manage multiple applications, as they allow operators and users to specify their own grouping, and to quickly navigate to a specific object, based on a local understanding of how the scheduler is used.

See Also

- Organizing Processes
- Creating Applications

Chapter 6. Getting Started - Using SAP CPS

Use

The following pages provide basic information on the process scheduler as well as instructions of the basic tasks to help you get a running system for scheduling processes across SAP Systems in your landscape. If your environment has already been setup by an Administrator, you may skip the sections covering configuring and connecting to SAP Systems.

There are online tutorials available at the Redwood website [http://www.redwood.com/education/tutorials/-process-automation] that show you basic functionality. The first tutorial, *basic job definition*, will load automatically and guide you through the process of creating a job definition and submitting it. Most tutorials require a free registration and are marked with a lock icon, and we strongly encourage you to register. You can also take a look at the job chain definition [http://www.redwood.com/education/tutorials/process-automation/jobchain] and adding a process server [http://www.redwood.com/education/tutorials/process-automation/addingserver] tutorials, without registering. These tutorials are for all SAP CPS users, feel free to register; the reason why Redwood wants you to register is only to get data on who watches which tutorial in an effort to improve the overall experience.

You should be at ease with the software and able to integrate various SAP Systems as well as submit and monitor jobs. Additionally, you should be able to Import jobs from SAP Systems and tailor them to fit your needs.

Topic	**Description**
First Steps	This documentation contains an overview of logging on to SAP CPS systems, navigating on the user interface, personalizing user settings, and information about accessibility. Logging onNavigationPerforming Basic Actions
Working with Tools and Features	Submitting a Job or Job Chain Monitoring and Managing Jobs Configuring an SAP System for Central Process SchedulingConnecting to SAP SystemsCreating Isolation Groups and Assigning them to UsersImporting CCMS JobsCreating a Job Chain with Sequential JobsThe Job Definition TypesCreating and Editing Job Definitions

Logging on

Use

When you log on, you are put into an isolation group. If the value of the license key *System.sharedPartitions* is a numeric value, you are entitled to use one isolation group at any given time. You will have to choose for the session at logon if you are a member of several isolation groups.

Note

If you get an HTTP error `403 - Access Denied` error after logging in, your user may be lacking the `scheduler-user` role. If this is the case and you are unable to logout, you will have to restart your browser.

Version Information and Help

General Version information, manuals and tools can be found in the *Product Information* screen, which you can open by choosing the blue **i** icon.

Task	**Description**
Enter Registration Details	Register for tutorials
Setup Wizard	open setup wizard to connect to SAP or platform agents.
Enable/Disable Updates	Enables or disables the updater, which checks online for newer versions of SAP CPS
Get Support Files	Allows you to download log files; especially handy when you are experiencing issues with SAP CPS
Support	The support jsp allows you minimal system access in case you can no longer access SAP CPS; Only access when SAP Support Services instruct you to do so.
Build:<xxxxx>	The exact version information of the currently deployed web application

Prerequisites

To access SAP CPS the following is required:

- a web browser
- a username/password with appropriate authorizations
- a URL

The URL has the following syntax *http://&lt;hostname&gt;:&lt;port&gt;/scheduler*

- <hostname> is the hostname of the SAP NetWeaver system where SAP CPS is installed.
- <port> is the HTTP port of the SAP NetWeaver system.

Note

If a pop-up blocker is installed, you should configure it to always allow pop-up windows from the URL of SAP CPS. Refer to the documentation of your pop-up blocker if you are unsure how to proceed.

Procedure

1. Enter the URL into the browser's address bar. A pop up window that requires a username and password to log-on with appears.
   a. If a pop-up blocker is installed, a blank page with *Please log in* gets loaded, choose *Please log in*.
2. Enter the username and password.
3. Choose *Log on*.

Navigation

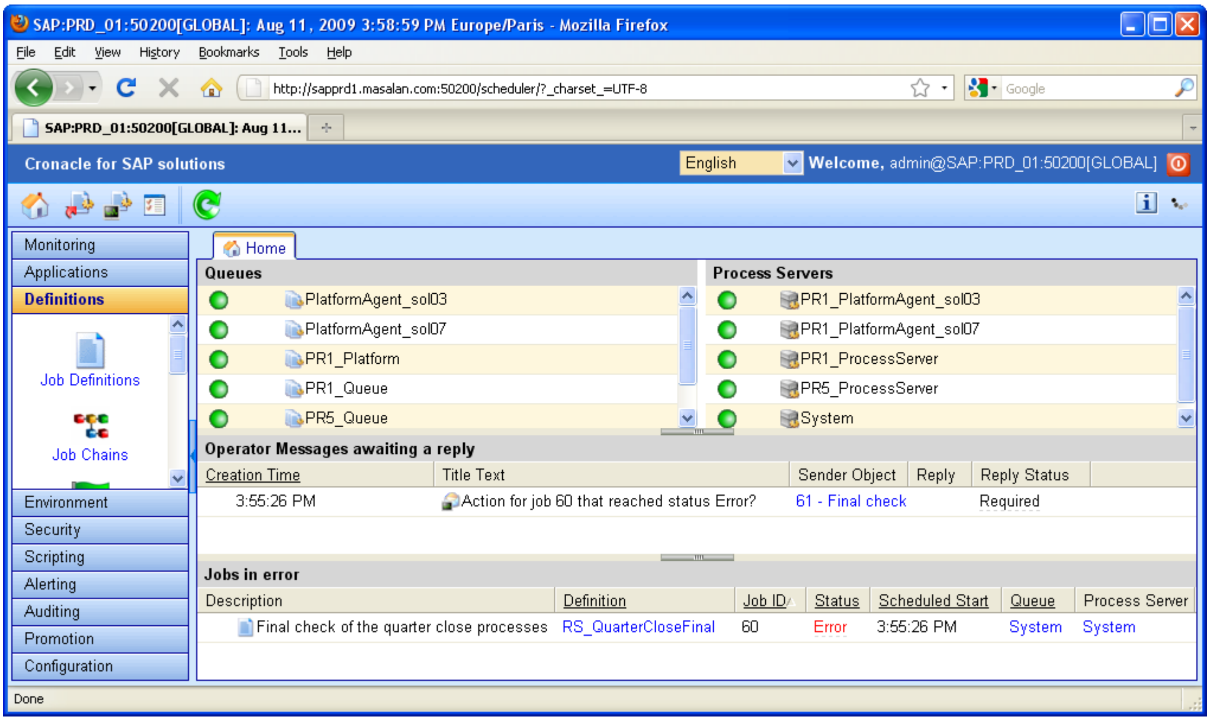
Use

In contrast to previous versions, like SAP Process Scheduling by Redwood version 7.0.4, you can access SAP Process Scheduling by Redwood only by its web-based UI, which has been adapted for screen-readers. This UI allows you to access the complete functionality for which you have authorizations. It offers the same features as an application installed on your local workstation, the browser context-menu has been replaced with a product specific context-menu.

The Main Screen

Upon successful log-in, the pop up window closes, and the main window refreshes to show the user interface. It may take a few seconds to reorganize the display if this is the first time. The home page shows the status of the entire system. This home page can be used to access other parts of the system.

The following figure shows the main screen.


SAP:PRD_01:50200[GLOBAL]: Aug 11, 2009 3:58:59 PM Europe/Paris - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://sapprd1.masalan.com:50200/scheduler/?_charset_=UTF-8
Google
SAP:PRD_01:50200[GLOBAL]: Aug 11...
Cronacle for SAP solutions
English
Welcome, admin@SAP:PRD_01:50200[GLOBAL]
Monitoring
Applications
Definitions
Job Definitions
Job Chains
Environment
Security
Scripting
Alerting
Auditing
Promotion
Configuration
Home
Done

Queues	Process Servers
PlatformAgent_sol03	PR1_PlatformAgent_sol03
PlatformAgent_sol07	PR1_PlatformAgent_sol07
PR1_Platform	PR1_ProcessServer
PR1_Queue	PR5_ProcessServer
PR5_Queue	System

Operator Messages awaiting a reply

Creation Time	Title Text	Sender Object	Reply	Reply Status
3:55:26 PM	Action for job 60 that reached status Error?	61 - Final check		Required

Jobs in error

Description	Definition	Job ID	Status	Scheduled Start	Queue	Process Server
Final check of the quarter close processes	RS_QuarterCloseFinal	60	Error	3:55:26 PM	System	System

The title bar at the top of the page provides information about the current session, the ability to change language, and (on the far right-hand side) a logout button. The tool-bar is below the title bar, it provides access to frequently used facilities. The navigation bar is on the left-hand side, this allows access to the different screens that are shown in the main pane on the right-hand side. More details about these are provided later.

Browsing and Working with Objects

You can use the navigation bar on the left-hand side to access different objects within the system.

1. Choose *Definitions*, this shows all the definitions.
2. Choose *Job Definitions*. A new tab opens on the right-hand side, showing a list of all job definitions in alphabetical order by name.
3. Choose the first job in the list. A preview pane appears. The preview pane shows more detailed information about the currently selected object in the table.
4. Select the *System_Info* job in the list. When this job definition is selected the preview pane changes to show its details.

5. In the details pane, choose the Name, a new tab is created with only that job definition; it is the show page. Now, use the context menu of *System_Info* in the table. A pop up menu appears.

6. Choose *Submit*. This opens the Submit Wizard. In the Submit Wizard times and dates at which the job runs can be specified, as well as the queue and priority to use. This job runs fine with the defaults, so choose *Submit* in the white box on the left-hand side.

7. Select *Show all jobs*, and press the submit button on the bottom row. This closes the Submit Wizard and the job monitor is displayed where the newly submitted job is displayed. If the system is busy choose *Refresh* from the context menu to see the new job.

8. Select the new job in the table, and move to the preview pane. Here information about the currently submitted job is displayed. Some items have a plus (+) in a box next to them, these can be expanded to show more information. The second item with a plus (+) next to is *Files*, this shows the files that the job created while it was running.

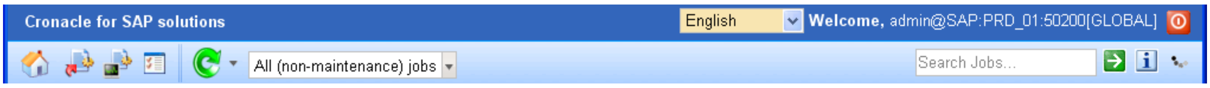
9. Choose plus (+) to expand this. This job created two files while it was running, a file containing the output and another containing eventual errors.

10. Choose the file name, this opens a new tab, showing the file contents. The file contains a list of *System Properties* and other information that is useful for SAP Support if there is a problem.

11. Choose the (x) on the left in the tab to close the tab (hover over the little icon), this takes you back to the job monitor.

The Title bar and Tool bar

The following figure shows the title and tool-bars.


Cronacle for SAP solutions
English
Welcome, admin@SAP:PRD_01:50200[GLOBAL]
All (non-maintenance) jobs
Search Jobs...

The title bar provides information about the current session and the ability to change the language and log out.

- To change the language - choose the new language from the drop-down box.
- To log out - choose the logout icon on the far right-hand side.

The tool-bar provides quick access to several frequently used facilities

- *Home* - the home page provides an overview of system status.
- *Submit job* - this link submits a new job.
- *Job Monitor* - this provides a direct link to the job monitor.
- *Options* - this opens the user specific settings dialog.
- *Refresh* - refreshes the currently selected tab.
- *Filter* - the context-sensitive filter is displayed in the job monitor by default, it can be displayed for all tabs that display objects but you need to create a filter first when there is none.
- *IntelliSearch* - quick and efficient object search, displayed for the current overview.
- *Information* - allows you to access version information, the online help as well as the link to register for tutorials and support.

The title-bar provides the following:

- *Language switcher* - allows you to change your current language, all open tabs are the reloaded with the new language.
- *Greeting* - in your chosen language, the greeting provides information about the logged on user and the system you have logged on to, as well as the current isolation groups in brackets [].
- *Log off* - on the upper right-hand corner there is a log off button.

Depending on the currently selected overview, more overview-specific icons can be displayed on the title-bar.

Actions

All objects have actions associated with them. In some cases not all actions are available.

There are three reasons why this may be:

- The local security policy
- Restrictions in the license.
- The object has become read-only, like a job that has reached the status *queued*.

There are 5 basic actions that are supported by most objects:

- *New* - create a new object
- *Edit* - edit an existing object
- *Delete* - delete an object
- *Export* - export the object to a CAR file for backup
- *Refresh* - refresh the current overview to display recent changes

Many objects also support specific actions. Some examples are:

- Jobs can be restarted, resubmitted, held, released, and so on.
- Job definitions and job chains can be submitted as jobs.
- Queues can be held and released
- Process servers can be started and stopped.

Some actions require further input before they can be completed, these actions are shown with a trailing ellipsis (...) to indicate that further input is required. Actions without the trailing ellipsis are executed immediately.

- Example actions that require further input: *Edit*, *Add Job Note* and *Resubmit*.
- Example actions that don't require further input: Hold, Release and Refresh

Additional actions for navigation:

- *Help* - display the help topic covering the current tab
- *Refresh* - update the display for the current tab

- *Close tab* - will close the current tab and switch to the previously selected tab.
- *Close all other tabs* - closes all tabs except for the current tab

Restrictions on Actions

Licence restrictions can prevent the following actions:

- Creating job definitions -in this case only standard job definitions or ones imported from CCMS are available.
- Editing or Duplicating job definitions of type `System` - these are supplied by Redwood.
- Creating job definition types - these are supplied by Redwood.
- Creating Users or Roles - these are automatically synchronized with SAP UME.

Furthermore, it is not possible to create or delete time zones, as these are fixed. It is possible to change the description of a time zone.

Navigating with Screen Readers

Use

Means of navigation have been improved for screen-reader users. This topic explains what actually changed and how to use it.

Basic flow

The main navigation control groups links for all object types. The control will only show one group at a time with all links for the group. All links on hidden groups are not shown.

Examples of groups are:

- *Monitoring*
- *Definitions*

When a link of a group is followed, a tab page is opened, which consists of two visible parts:

1. An entry is added to a list containing the names for all currently open tab pages.
2. A page is opened (using a frame) containing the information associated with the object type.

Most of the pages opened are so called "overviews", which are lists of objects of a certain type. When a row in the list of such a page is chosen, a new subordinate page will be displayed. The page contains all attributes of the selected object and offers all possible actions on the object.

Extra Controls on the Main Page

Besides the controls described in the basic flow, the main page contains more controls.

1. language selector - This is a simple combo input, allowing to choose a language. Note that this now also influences the screen-reader.
2. log-out button - Simple button, performing a log out.

3. tool-bar - Contains action like: show homepage, submit job, show job monitor page, show options dialog. It will also contain context-specific actions for the current page.
4. search jobs - Input used for using the intellisearch function.
5. information dialog - Shows the systems version information.
6. homepage - This is the default page shown after logging on. It contains general information about the system.

Using the Screen-Reader for Basic Flow

Groups can be accessed since they are links. The tool-bar also contains links and they are placed higher up on the page, so they will appear first. Selecting a group will expand it, like "Monitoring", after that it will show the links inside, these are prefixed with the "open" keyword. Selecting a link will open a page.

At anytime one page is shown, however more can be open at the same time, with the exception of detail pages. It works like tab pages, selecting a tab representing the page will show it. To display a page, you can make use of the fact that the tab names are level 1 headings, and can be navigated as such, by using the '1' key.

Again, most pages contain a list of objects, by navigating through level 2 heading, one navigates through the rows, *arrow down* can be used to announce row contents. After selecting a row, a detail page is displayed besides the overview. With level heading 3 navigation you can directly navigate to it. From here you should be able to access all the data and actions of the object.

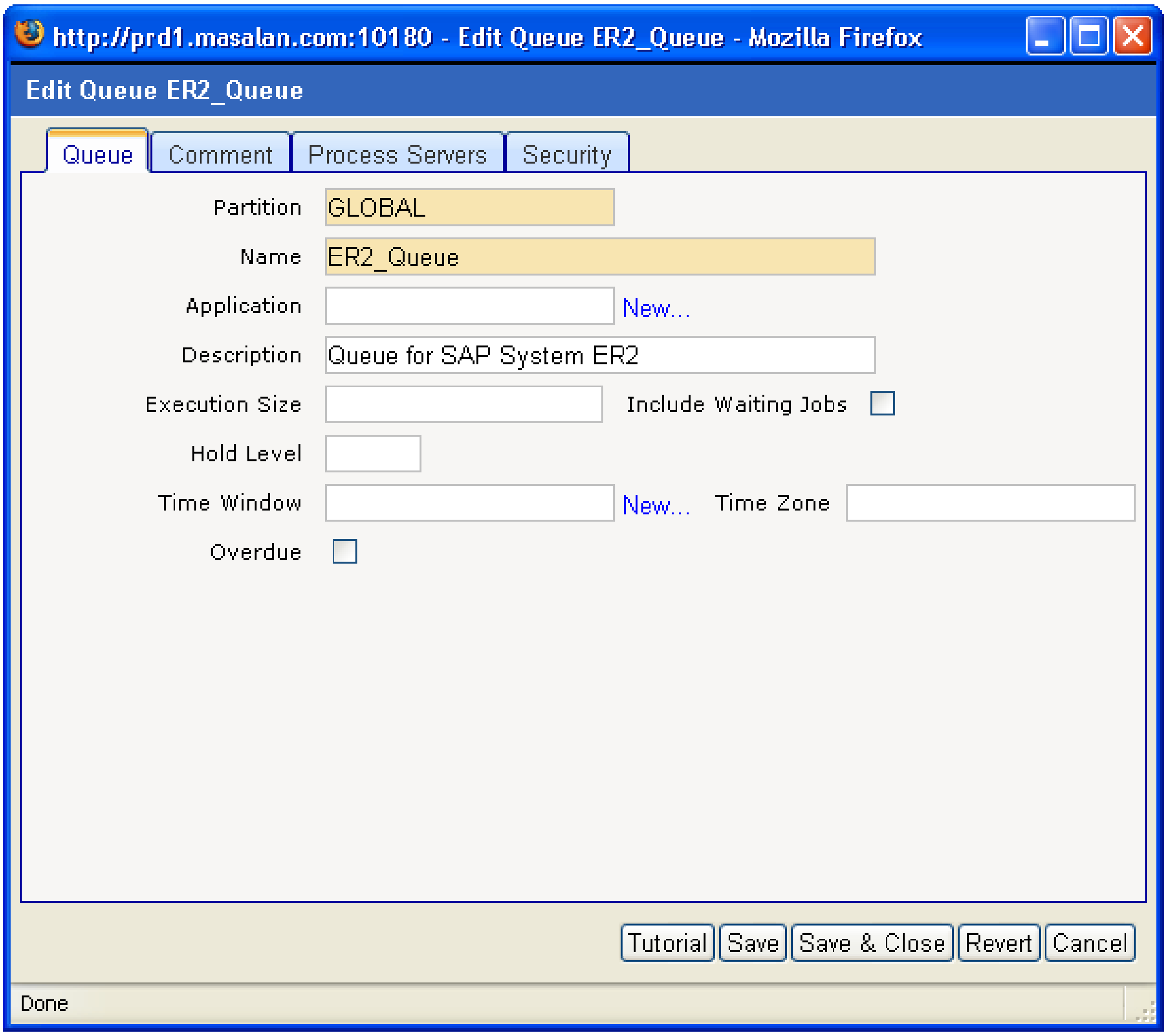
Performing Basic Actions

Use

The Basic Actions that most objects support are create, edit, delete, export, and refresh.

Create and edit windows are pop-up windows that can be used to create or change an object's details. A simple Edit window (for the *Queue* object) is shown below.

The following figure shows the first tab of the queue editor.



http://prd1.masalan.com:10180 - Edit Queue ER2_Queue - Mozilla Firefox

Edit Queue ER2_Queue

Partition	GLOBAL
Name	ER2_Queue
Application	
Description	Queue for SAP System ER2
Execution Size	
Include Waiting Jobs	□
Hold Level	
Time Window	
Time Zone	
Overdue	□

Done

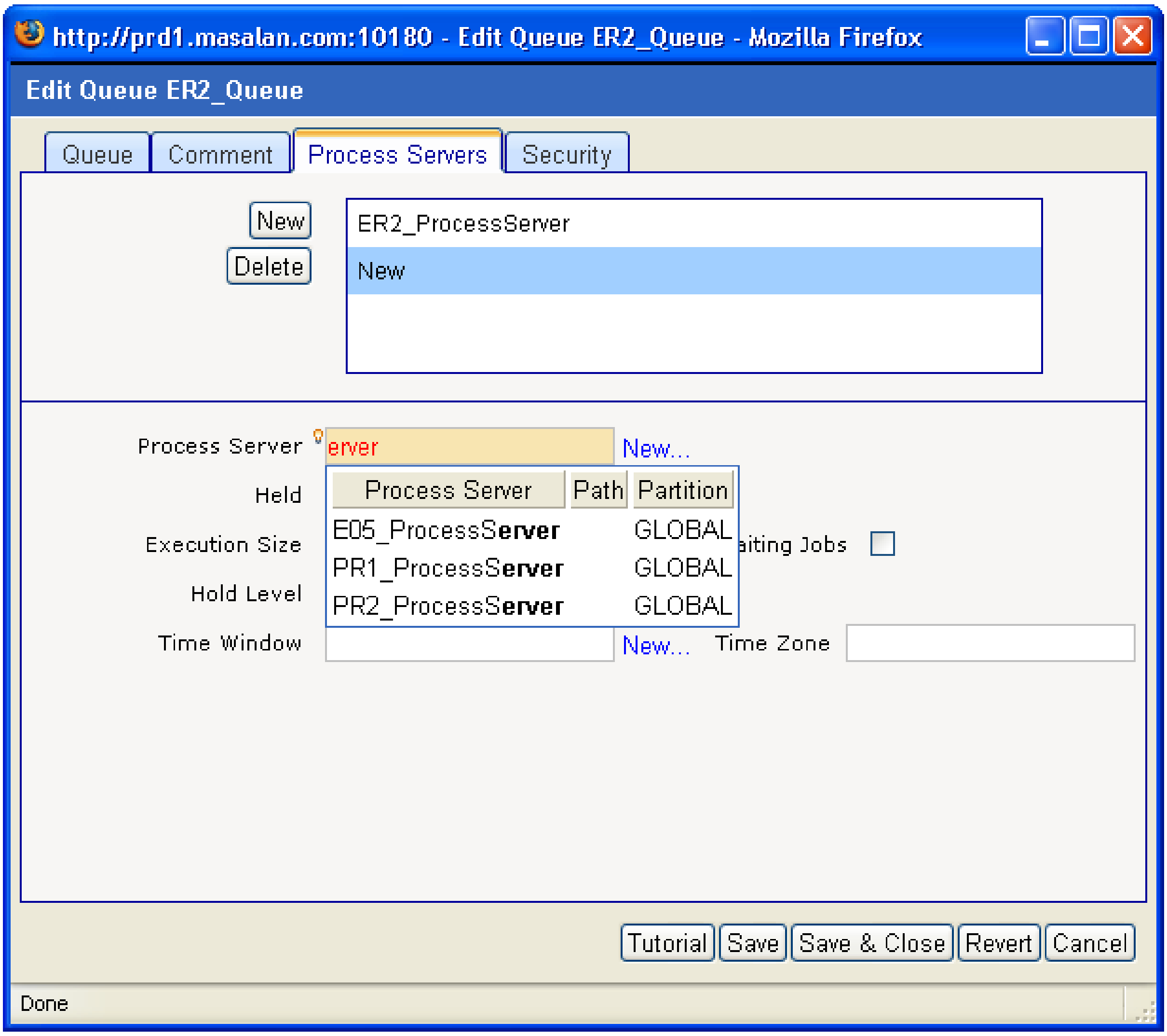
The queue editor has several tabs on it, the *Queue* tab can be used to edit the main object's own details. The *Process Servers* tab is used to add or remove the process servers fed by the queue.

Fields with a darker background are mandatory. Fields with a white background are optional. Certain fields have an auto-complete functionality. When these fields are chosen a small light bulb is shown next to the field on the left-hand side. As soon as one or more letters have been entered into such a field, a drop-down provides possible choices and automatically filters these to match the letter sequence entered. The letter sequence does not have to start with the first letter of the object's name, for example, in a time window field, *Ind* as well as *dence* both match the time window *System_US_IndependenceDay*. You can use the wildcards % (any numbers of characters) as well as _ (single character) and can use the escape character \ to escape these wildcards.

Some fields also have a *New* link next to them. This link can be used to create a new object of the appropriate type in a separate window while the first object is edited.

To create a new relationship, choose *New*. This populates the lower window with fields for the object's relationships.

The following figure shows the second tab of the queue editor, with several relationships.



http://prd1.masalan.com:10180 - Edit Queue ER2_Queue - Mozilla Firefox

Edit Queue ER2_Queue

ER2_ProcessServer

New

Process Server: erver

Process Server	Path	Partition
E05_ProcessServer		GLOBAL
PR1_ProcessServer		GLOBAL
PR2_ProcessServer		GLOBAL

To navigate between different relationships, use the white box located above.

Security

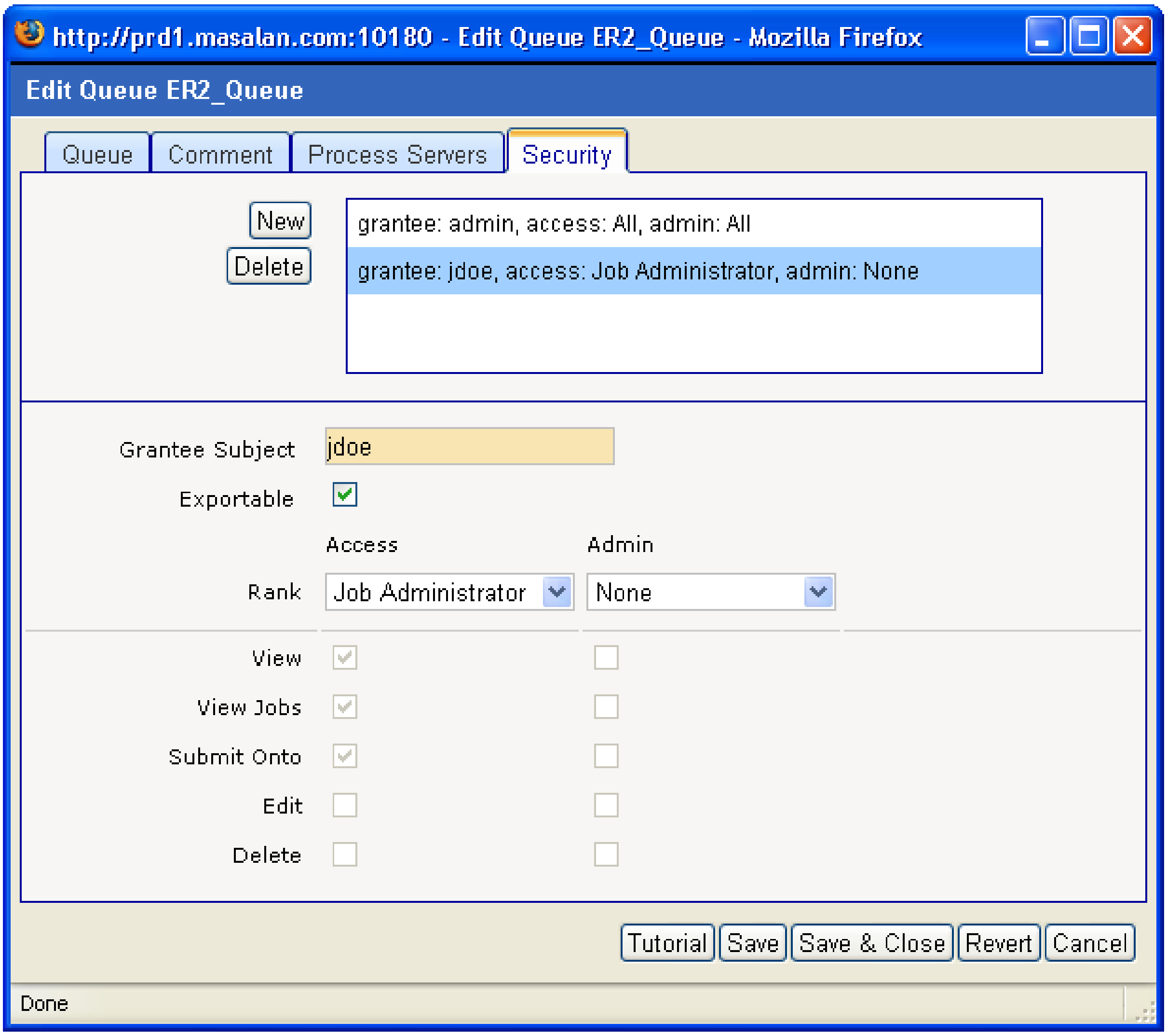
In the Free of Charge version, you can grant one or more of the built-in roles to your custom roles, which allows you to manage security in a quick and easy way. It has limitations, as the built-in roles are task-oriented and no object-specific privileges are available. Your SAP CPS is in isolation mode, which means that all SAP Systems are isolated from each other. You choose an isolation group when you log on which allows you to interact with one SAP System only.

Note

Custom roles are created in your authentication system and are created in SAP CPS as soon as a user with the role successfully logs on.

When you have the Security Module, which requires the *Module.Security* license key and is part of the Chargeable version, an additional *Security* tab is displayed and allows you to grant specific users or roles object-level access to the object you are editing. This allows fine-grained security. You can also grant privileges of object types to your custom roles on the following levels:

- System – applies to all partitions.
- Isolation group wide – applies to all partitions in an isolation group.
- Partition – applies to a single partition.


http://prd1.masalan.com:10180 - Edit Queue ER2_Queue - Mozilla Firefox

Edit Queue ER2_Queue

grantee: admin, access: All, admin: All

grantee: jdoe, access: Job Administrator, admin: None

Grantee Subject: jdoe

Exportable ☑

	Access	Admin
Rank	Job Administrator	None
View	☑	□
View Jobs	☑	□
Submit Onto	☑	□
Edit	□	□
Delete	□	□

You cannot grant privileges directly, you grant ranks of privileges. For example, if you grant the *Edit* rank to a custom role, the users that have this role have the *View* and *Edit* privileges. This reduces the amount of grants you need to make and avoids human error, as you cannot grant the *Edit* privilege without the *View* privilege, which would not make sense. You can of course grant multiple ranks, like *Edit* and *Delete*, to your custom roles.

Procedure

To create an object of a particular type:

1. Find the object type in the navigation bar.
2. Choose the icon to go to the overview screen for that object.
3. Choose *New* from the context menu.

To edit an object of a particular type:

1. Navigate to the overview screen for that object.
2. Use the context menu of the object and choose 'Edit'.

To delete an object of a particular type:

1. Navigate to the overview screen for that object.
2. Choose the object that is to be deleted in the table.
3. Choose *Delete* from the context menu.

Locating Objects Quickly

Use

The following features allow you to locate objects quicker:

- filters - a static filtering tool available in most object overviews; you only specify your search criteria once
- applications - you can navigate to the application the object belongs to
- intelliSearch - a dynamic filtering tool available in most object overviews; you need to specify your query each time

If you want to quickly locate objects, and the objects are not always the same, use intelliSearch instead of filters. IntelliSearch allows you to quickly display one or more objects by simply entering specific data, like a distinct subset of a name. See the intelliSearch topic for more general information on intelliSearch or the object-specific documentation. You reach the object-specific documentation by navigating to the object's overview and hitting the *F1* key.

You might already be familiar with filters from the job monitor, most objects can be filtered by name, application and other object properties.

You may specify any name for a filter, however, some words like *System* and *SAP* are reserved words, you can use *Sysdm* or *S@P* instead, for example. Note that the interface will display the description instead of the name of the filter when it is set, and that there are no reserved words for the description. Filters can also be used to generate reports, if you have the reporting module.

When you define a filter, you define a SQL query, SAP CPS makes it easy for you to define such a query by providing you with a list of properties as follows, for each property you can *include* or *exclude* objects.

The following was taken from job definition filters

```
Where the name contains specific keywords
Where the name starts with a prefix
Where the Job Definition belongs to an application
Where the Job Definition belongs to an application containing specific keywords
Where the Job Definition is defined as top level
Where the Job Definition is a built-in one
```

Procedure

Creating an object filter:

1. Navigate to *Definitions > Job Definitions*
2. Choose *Filter > Create Filter* from the context menu.
3. Fill in the name and description fields
4. Choose one or more criteria.
5. Choose *Save & Close*

Result

The new filter will be selected and only objects that match your filter will be displayed.

Example

Creating a job definition filter for system job definitions:

1. Navigate to *Definitions > Job Definitions*.
2. Choose *Filter > Create Filter* from the context menu.
3. Fill `sysdm` in to the name field and `System Job Definitions` into the description field.
4. Choose *Where the name contains specific keywords*, then *specific keywords* from *Selection* field below.
5. Fill in `System`, choose *Add*, then *OK*.
6. Choose *Where the Job Definition is a built-in one*. This is to illustrate that you can select multiple conditions, `System` is a reserved word and only built-in definitions may have it in their names.
7. Choose *Save & Close*.

Advanced Object Search

Use

SAP CPS allows you to search seamlessly through objects with the new search box, which is now available for most objects. The search only searches in the current tab and by default searches for the name on most objects. For some objects like operator messages, for example, you would rather search for the *Title text*, which is the title of the operator message. You can use prefixes to search on object-specific properties, like `lmt`, which stands for *Last Modification Time*. The search is *case-insensitive*. Default objects often have translated field values, you should always search in your session-language.

You can search on multiple properties by just appending a new prefix and search criteria. Depending on the amount of objects which match your criteria, one or more will be listed.

The following prefixes are available for most objects:

Prefix	Description

n, name	Name field of the object
d, desc, description	Description field of object
c, com, comment	Comment field of object
a, app, application	Application field on object

The `n` or `name` prefix is optional, all other prefixes are required, so if you want to search on the name of the object, you just enter your search-phrase.

Note

You can only use valid prefixes. Not all objects have a *Name* property, like operator messages for example, in this case the `n` prefix is not available. The available prefixes are documented in the topic covering the object. Press the `F1` key to access context-sensitive help with the table of available prefixes.

Syntax

```
<prefix>:<search_criteria> <prefix>:<search_criteria> ...
```

Note

There is no space between the *:* and the prefix or search-phrase.

Note

If the search-phrase contains spaces, the search-phrase must be surrounded by double quotes (").

If you just enter the search-phrase, the search will be carried out on the name of the object:

```
<search_criteria>
```

Some objects do not have a *name* prefix, in these cases the default prefixes have been mapped to object-specific prefixes.

Example

To search for a job definition with the description *Import Java Archive (JAR) into a Library*:

1. Navigate to *Definitions* > *Job Definitions*

2. Enter the following text into the *Search Job Definitions* box to display only jobs with *import* in the description field:

```
description:import
```

In this example, too many job definitions matched the search! More search criteria needs to be added to refine the search:

```
description:"import Java" application:SystEm
```

Advanced Actions

Use

The following table lists the topics that describe frequently used tools and features.

Topic	Description
Working with Tools and Features	Submitting a Job or Job Chain Monitoring and Managing Jobs Configuring an SAP System for Central Process SchedulingConnecting to SAP SystemsCreating Isolation Groups and Assigning them to UsersImporting CCMS JobsCreating a Job Chain with Sequential JobsThe Job Definition TypesCreating and Editing Job Definitions

Editing Objects in XML

Use

You can edit objects in XML in the user interface, which allows you to use an external editor and perform a global search and replace, for example. To enable XML editing, you need to set the *Allow edit as XML* option in the Setting User Parameters.When you export objects, they are saved as XML files in CAR files, see Importing SAP CPS Objects for more information.

You can also expand exported car files, and edit the files. The car files are in fact jar files, which can be opened by programs understanding zip. If you do not use `jar`, the Java executable to create jar files, to re-assemble the car file, you will not be able to use the embedded upload functionality found in car files.

Special care needs to be taken when you edit objects in XML.

- Special characters must be escaped using entities
- No trailing spaces are allowed

Special Characters

According to the W3C Recommendation [http://www.w3.org/TR/REC-xml/#syntax], special characters that must be escaped include the following:

`<, >, ", &`

The following table lists the characters that have to be escaped and the entities you may use instead.

Character	HTML Entity	Decimal Entity
<	&lt;	&amp;#60;
>	&gt;	&amp;#62;
"	&quot;	&amp;#34;
&	&amp;	&amp;#38;
'	&apos;	&amp;#39;

Note

When you use other entities, these are replaced with the actual characters once you save by the XML parser. If you need to use character entities, you should always use decimal entities wherever possible.

The above HTML entities are recognized in XML, but others might not be and will prevent you from saving the object.

Example

The following example illustrates what **not** to do:

```
<Description>My description contains a trailing space, which is not allowed! </Description>
<Comment>I must remember not to use special characters like <, which needs to be escaped as < or &lt;!</Comme
```

See Also

Importing Objects from Remote CAR Files

Chapter 7. Administering the Scheduling Landscape

Use

These main administration tasks for fine tuning the security policy or tuning job execution cover the whole Scheduling Landscape, both the Central and all satellite systems. SAP CPS offers the possibility to schedule over multiple systems (load balancing), it also integrates with a lot of the external non-SAP applications. Process servers run natively on Unix, Windows and other platforms when you have appropriate licenses.

1. Checking Your License
2. Configuring the Central and Satellite Systems
    a. Understanding the Different XBP Interface Releases
    b. Enhanced SAP Interfaces with Redwood Transports
    c. Configuring an SAP System for Central Process Scheduling
    d. Connecting to SAP Systems
3. Managing Roles and Users
    a. Isolating Users and Objects
    b. Security in SAP NetWeaver
    c. Creating Isolation Groups and Assigning them to Users
    d. Assign Users from External Platforms to Jobs
    e. Privileges
4. Auditing Object Changes
5. Generating Reports
6. Setting User Parameters
7. The Registry
8. Scaling the Scheduling Resources on the Central System
    a. Using Process Servers
    b. Using Queues
    c. Prevent Simultaneous Execution of Processes
    d. Creating Event-Driven Schedules
    e. Tuning SAP CPS
    f. Balancing the Load

9. Organizing Processes

    a. Creating Applications

10. Importing and Exporting SAP CPS Objects

11. Managing Job Output Formats

12. Regular Activities for SAP CPS

Checking Your License

Use

A license enables certain licensed features, features that require a license item include:

- Scripting module
- Mail module
- SNMP module
- Import/Export module
- Expiration date
- Number of SAP Systems
- Number of platform agents

A license is made of license keys that have a value, the value can be a date, number, boolean or string.

When you lack a license key, that license key will automatically have the value of false.

Navigate to *Configuration* > *License Keys* to inspect your license.

Your license should look similar to the one below:

```
Module.Scripting                         true
Module.ImportExport                   true
ProcessServerService.External.limit      100
ProcessServerService.OS.limit           100
ProcessServerService.SAP.limit     2
company.contract                   SLA-MSLN-PROD-1
company.name                           Masalan Inc
product.expiry                           20100630
request.requesting_version           M37
signature  1;a300a310a300600a318a319a320a321a330a330400a330200a330600a340a340400a340600a350a380a380800
system.id                           prd1
```

Note

In the above license, the licensee can create 100 external schedulers in total, that could mean 100 OS process servers and no SAP process servers or 98 OS process servers and 2 SAP process servers, for example. It does *not* mean that the licensee can create 102 (100 OS and 2 SAP) process servers.

Warning

Reverting from a Chargeable to a Free of Charge version is **not supported**. This is due to the fact that there are restrictions in the Free of Charge version which do not exist in the Chargeable version.

The following table illustrates some of the license keys that are available.

License key	Description
company.contract	Contract number.
Module.Scripting	Scripting using RedwoodScript.
Module.Security	The security module allows individual grants for system and object privileges.
Module.ImportExport	Import/export repository objects.
Module.Mail	Use the System_ConfigureMail and System_SendMail jobs to send email.
Module.SNMP	Use the System_SendTrap job to send SNMP traps.
ProcessServerService.External.limit	Total number of external process servers (for example OS and SAP process servers)
ProcessServerService.OS.limit	Number of platform agent on UNIX/Windows.
ProcessServerService.SAP.ApplicationsAPI	The SAP Applications API is available
ProcessServerService.SAP.cc	Closing Cockpit features are available.
ProcessServerService.SAP.CalendarSync	SAP Calendar synchronization is available
ProcessServerService.SAP.MassActivities	Mass activity scheduling is available
ProcessServerService.SAP.ProcessChainRestart	Process chain restart is available
ProcessServerService.SAP.SolutionManager	Solution manager integration is available
ProcessServerService.SAP.XBPAllVersions	Use the Redwood transports that allow full functionality in all XBP versions.
ProcessServerService.SAP.XBPExternal	Use the SAP external command jobs.
ProcessServerService.SAP.XBPVariant	Support for working with variants using our user interface.
ProcessServerService.SAP.limit	Number of SAP process servers/SAP Systems that can be created, `2` in this example. If the value is `0`, then you can connect to an unlimited amount of SAP Systems, however, you can only interact with one at any given time, see the license restrictions below.

Note

When you install a new license from a different contract, the old license keys remain in place and have no effect. You can use the license manager to remove old licenses by deactivating the contract. The license manager is available at: `http://server:port/scheduler/ui/license/manager.jsp`.

Searching License Keys

You can search for license keys using the *Search License Keys* box located under your username on the top right-hand side of the user interface. This is known as intelliSearch and allows you to specify complex queries in a simple way using prefixes. Prefixes are used to specify which property you are searching in and have short and long syntaxes. For example, if you want to display all license keys for contract `SGO-FR-0210-008-P` you would use the following search criteria:

```
c:SGO-FR-0210-008-P
```

You can search more than one property, as follows:

```
c:SGO-FR-0210-008-P n:Module.SNMP
```

Note

No spaces should be entered before or after the colon (:).

See the Advanced Object Search for more information.

The following table illustrates the available prefixes for license keys:

Prefix	Description
n, name	name of the license key (case sensitive)
con, contract	contract name (case sensitive)

Special Restrictions

License Expiration

When a license has an expiration date, usually for testing purposes, and it expires, all scheduled jobs will continue to run and be rescheduled by their submit frame, if applicable. You will, however, not be able to access the system via its user interface. If you are in this situation and you want to stop SAP CPS, you need to stop the SAP CPS application running in the application server. How you do this depends on your application server. You might wish to remove SAP CPS completely. See the topic for more information.

SAP Systems

Note

You may be restricted to connecting to a single SAP System from within SAP CPS at any given time; this depends on your license, your privileges and the value of the *ProcessServerService.SAP.limit* license key.

If the *ProcessServerService.SAP.limit* license key is limited to `0`, the following restrictions are in place:

- one SAP System may be created per partition with one XBP client.
- any partition with a `null` isolation group is read-only.
- only the system can create partitions with a `null` isolation group.
- all monitors from SAP (via XAL) will be created in the appropriate isolation group.
- SAP batch event will raise events in the isolation group only.
- job control rules will function only for jobs from the defined XBP client, child jobs will be picked up regardless in which client they were created.
- variant management will be limited to the defined XBP client.

- BW restart will be possible, however, it requires the *ProcessServerService.SAP.ProcessChainRestart* license key
- Mass activities will be possible, however, they require the *ProcessServerService.SAP.MassActivities* license key
- Redwood transports will be available for mass activities
- SAP Applications API will be available, however, they require the *ProcessServerService.SAP.ApplicationsAPI* license key
- *SAP_AbapRun*, *SAP_BW_ProcessChainRun*, and all the other SAP job definitions that run jobs will only allow ABAP reports, variants, and process chains that have been imported into SAP CPS

Process Servers

The maximum amount of external process servers is specified by the *ProcessServerService.External.limit* license key, it would normally be coupled with two other license keys that share the same number which are *ProcessServerService.OS.limit* and *ProcessServerService.SAP.limit*. To allow you the greatest flexibility, SAP allows you to create as many OS and SAP process servers as you want, as long as the grand total `OS process servers + SAP process servers` does not exceed the value defined in *ProcessServerService.External.limit*. At any time, you may shutdown and delete a process server to make room for another, regardless of either being an SAP or OS process server. Note, however, that to delete a process server all dependent objects, such as jobs that ran on that process server, need to be deleted first.

Vendor Restrictions

Not all vendors distribute the same functionality, please see the Different Versions section for more information.

See Also

- Different Versions

Different Versions

Use

Depending on the vendor and requirements, two versions of SAP CPS are available. The Free of Charge version is only available to SAP customers. The Chargeable version can be purchased from all vendors. Additional modules are available to complement the features of the Chargeable version, however, some modules can only be purchased from Redwood Software and not from other vendors.

The Free of Charge version

The following license keys are available in the Free of Charge version.

License key	Description
System.NamedUsers.limit=10	Limit is for 10 SAP CPS users.
System.Production=true	The Free of Charge component is able to connect to production SAP systems.
ProcessServerService.SAP.XBPExternal=true	Allow users to run SAP initiated external commands through SAP CPS rather than through SAP itself.

ProcessServerService.SAP.ApplicationsAPI=true	Provides integration for SAP applications that have been written to take advantage of the SAP specific API direct into SAP CPS.
ProcessServerService.SAP.SolutionManager=true	Allows integration with change control and documentation management aspects of SAP Solution Manager.
ProcessServerService.SAP.limit=0	Set to 0 in Free of Charge version to enforce isolation mode. Allows individual management of unlimited SAP systems but no cross SAP system or client scheduling.

The Chargeable version

The following license keys are available in the Chargeable version.

License key	**Description**
ProcessServerService.SAP.limit=nn	Customer specific. Allows cross SAP system scheduling for `nn` SAP systems. Maintains isolation mode for any SAP systems beyond `nn`. Note this is set to `0` in the Free of Charge version.
ProcessServerService.OS.limit=nn	Customer specific. Allows OS job management and file events on `nn` Windows or UNIX systems. Note that SAP sells agents for these platforms only. See the *Operating System Platform specific* section below for other OS platforms.
ProcessServerService.External.limit=nn	Imposes a total Process Server count made up from a combination of SAP and UNIX or Windows systems.
ProcessServerService.SAP.XBPExternal=true	Allow user to run SAP initiated external commands through SAP CPS rather than through SAP itself.
ProcessServerService.SAP.ApplicationsAPI=true	Provides integration for SAP applications that have been written to take advantage of the SAP specific API direct into SAP CPS.
ProcessServerService.SAP.SolutionManager=true	Allows integration with change control and documentation management aspects of SAP Solution Manager.
ProcessServerService.SAP.cc=true	Allows integration with SAP Financial Closing Cockpit as part of SAP Fast Close solution. Requires purchase of SAP Financial Closing Cockpit (available from SAP AG).
ProcessServerService.SAP.CalendarSync=true	Allows import of SAP calendars into SAP CPS to aid migration.
ProcessServerService.SAP.ProcessChainRestart=true	Allows automated restart of SAP NetWeaver BI process chains.
ProcessServerService.SAP.XBPAllVersions=true	Allows use of Redwood transports to provide full XBP functionality across all versions of XBP in use.
ProcessServerService.SAP.XBPVariant=true	Enables ABAP job variants to be managed from within SAP CPS instead of within SAP.
ProcessServerService.SAP.MassActivities=true	Allows management of Mass Activity Processing (for example with IS-U).
System.NamedUsers.limit=unlimited	Allow unlimited SAP CPS users. Note this is set to `10` in the Free of Charge version.
System.Production=true	Allow connect to production SAP systems.
Module.ImportExport=true	Allows migration of definitions between repositories (for example, migrating objects from dev > test > production).
Module.LoadBalancing= true	Enable SAP CPS load balancing capabilities

Module.Mail=true	Allows use of predefined job definitions to send email for basic alerting. Does not include Advanced Alerting Enhancement Pack.
Module.RemoteEvent=true	Remote event module that allows the jevent tool to be used.
Module.Scripting=true	Allows additional scripting functionality in REL (Redwood Expression Language). Allows use of Java for complex scripting.
Module.SNMP=true	Allows use of predefined job definitions to send SNMP traps for basic alerting. Does not provide capabilities of Advanced Alerting Enhancement Pack.

Additional Features & Modules

Advanced Modules

The following modules extend features already present in the Chargeable version.

License key	Description
Module.Alerting=true	Enables Advanced Alerting capabilities.
Module.Auditing=true	Allows tracking of changes to objects and restoration of objects from a previous iteration.

Business Automation Related

The following features extend SAP integration of the Chargeable version.

License key	Description
ProcessServerService.SAP.ApplicationsAPI=true	Enables integration for SAP applications that have been written to take advantage of the SAP specific API direct into SAP CPS. Provided with both Free of Charge and Chargeable versions.
ProcessServerService.SAP.cc=true	Only included with the Chargeable version of SAP CPS. Allows integration with SAP Financial Closing Cockpit as part of SAP Fast Close solution. Requires purchase of SAP Financial Closing Cockpit (available from SAP AG).
ProcessServerService.SAP.SolutionManager=true	Included with both the Free of Charge and Chargeable versions of SAP CPS. Allows integration with change control and documentation management aspects of SAP Solution Manager.

Operating System Platform specific

These features are *not* available from SAP AG and can only be purchased from Redwood Software. In any case, if you need the following functionality, please contact your account representative.

License key	Description
ProcessServerService.AS400.limit=nn	Defines number of IBM AS/400 (also known as iSeries or System i5) systems that can be managed by SAP CPS.
ProcessServerService.JCL_FTP=true	Allows JES jobs to be run via ftp using Windows/Unix platform agent.
ProcessServerService.JCL.limit=nn	Allows JES jobs to be run directly on the mainframe, SAP CPS software needs to be installed on the mainframe.
ProcessServerService.VMS.limit=nn	Defines number of OpenVMS systems that can be managed by SAP CPS.

See Also

- Checking Your License

Configuring the Central and Satellite Systems

Use

The SAP module allows you to connect to SAP Systems, and provides the default SAP job definitions.

SAP CPS requires an RFC user on every SAP AS (ABAP) system with specific privileges and connection details which are stored in *SAP Systems* repository objects.

- Setting up XBP 2.0 functionality on SAP Systems offering XBP 2.0.
- Setting up an SAP RFC user with the necessary privileges and connecting to SAP Systems.
- Using the default SAP job definitions
- Setting up Message Driven Beans Scheduling

The Free of Charge license uses only the built-in functionality of the interfaces of the connected SAP release; the Chargeable SAP CPS takes advantage of so-called Redwood transport files that enhance the functionality of the following interfaces:

- XBP 0.1 to 2.0
- Industry Solutions (IS) mass activities

For more information on the transport files, see Enhanced SAP Interfaces with Redwood Transports.

Note

If you are using an isolation group, you will be restricted from accessing more than one SAP System if the license key *ProcessServerService.SAP.limit* is set to `0`.

1. Understanding The Different XBP Interface Releases
2. Enhanced SAP Interfaces with Redwood Transports
3. Configuring an SAP System for Central Process Scheduling
    a. Enabling XBP 2.0 features
4. Connecting to SAP Systems
    a. Intercepting SAP Jobs
    b. Controlling Batch Input Sessions
    c. Interacting with SAP Events
    d. Parsing Log Files to set Job Status of SAP Jobs
    e. Integrating SAP Monitors

f. Sending Monitoring Data to SAP Systems

g. Controlling Communication Channels with SAP CPS via PI/XI

h. Monitoring SAP CPS Availability in CEN

i. Configuring the Message Driven Beans Scheduling Interface

5. Creating a SAP CPS RFC Server

a. Integrating Into Closing Cockpit

Understanding the Different XBP Interface Releases

Use

The BC-XBP interface, which allows external process schedulers access to CCMS Background Processing components of SAP Systems, has gone through three major upgrades. The releases are named after their version numbers: BC-XBP 0.1, BC-XBP 1.0, BC-XBP 2.0 and BC-XBP 3.0. All four revisions of the interface require certification.

Since the BC-XBP 3.0 interface is quite complicated we will follow the course of history and discuss the oldest interface first and then present the additional functionality of the later versions.

BC-XBP 0.1

The BC-XBP 0.1 release is strictly tied to SAP Basis releases 3.1 and 4.0. The BC-XBP 0.1 interface consists of a set of wrappers around some of the internal calls that were already supplied to ABAP programmers, and which are also used in the transactions that allow end users to submit background jobs.

BC-XBP 0.1 allows an external scheduler to:

- Create a job or modify a job and most of its attributes.
- Add, change and remove steps from jobs.
- Release a job and request it to be run now or as-soon-as-possible.
- Retrieve a job's definition, status, log, and spool files.

Customers that implemented interfaces using BC-XBP 0.1 quickly found that there were some minor issues with the functionality. Not all print parameters could be set using the external scheduler, and the job class was fixed to "C". Also, jobs could not be deleted by the external scheduler. Although no calls changed between Basis 3.1 and 4.0, some interfaces had different record structures. The most important change was the increase of the maximum job name from 8 characters in 3.1 to 40 in 4.0, and it was made possible to set the job recipient.

With BC-XBP 0.1 it was possible to schedule jobs from the external scheduler, but the SAP System remained leading since the scheduler could not even delete the jobs it started when it deemed these jobs should be deleted.

BC-XBP 1.0

The next version of the BC-XBP interface was released with SAP Basis 4.5.

The BC-XBP 1.0 version fixed most of the problems that were found with the 0.1 release, as it was now possible to delete jobs and set more print parameters (but still not all). Unfortunately it was still not possible to set the job class, and setting the job recipient was no longer possible.

BC-XBP 2.0

The BC-XBP 1.0 interface lived on in early releases of SAP Web Application Server with Basis releases 6.10 and 6.20. Support packages were quickly introduced that upgraded 6.10 and 6.20 to BC-XBP 2.0. As of early 2004, all SAP versions that are in "normal support" status have Support Packages that include BC-XBP 2.0. Thus, if you run release 4.6B or higher, and have the latest Support Packages installed, you have BC-XBP 2.0!

Note

You can verify that you have BC-XBP 2.0 with transaction **SE38**. If you can find an ABAP program named `INITXBP2` then you have at least BC-XBP 2.0 installed. An important step was made by SAP with BC-XBP 2.0. For the first time, the interface contained functionality that cannot be obtained from within SAPGUI. The CCMS scheduler by default runs with some disabled features.

Interception

The most important feature that was added with BC-XBP 2.0 is interception. The issue that it solves is the problem related to jobs that are scheduled in SAP directly and that are not under the control of an external scheduler. These jobs would then take up work processes that were meant for scheduled batch jobs.

The intercept concept only works in the presence of an external scheduler, and needs to be explicitly turned on. When turned on, the CCMS scheduler does some extra processing when a job's scheduled running time arrives. If interception is switched on, and the job was not scheduled by an external scheduler (for example SAP CPS), and the job is supposed to run now, then the job client, name and owner are compared to a list. If the job matches the list, it is not set to Ready but set to Scheduled and inserted into a list of 'freshly intercepted jobs'.

SAP CPS can then ask for this list of intercepted jobs and make its own decision on what to do with the jobs. As soon as SAP CPS has taken control of an intercepted job the SAP System will not report this job again the next time the scheduler asks for intercepted jobs.

Note

Jobs are not intercepted when they are created, they are intercepted when they are ready to run.

Since SAP was not able to introduce a new status in the batch job table for fear of breaking their own and other's applications, it had to re-use the Scheduled status in combination with the new intercept table. The result is that if an external scheduler asks for the status of an intercepted job it receives Intercepted, but the same job is Scheduled in tools that have not been modified. Since the online transactions are not modified, it means that any operator looking at the job with **SM37** will see that the job is 'back' in the Scheduled status. This was very confusing, and SAP has addressed the issue in AS ABAP 7.1. If you do not have this release, SAP recommend that all operators use the job monitor view to view the actual SAP job status.

There are three criteria that can be used to define which jobs should be intercepted by CCMS:

- Client.
- Job owner.
- Job name.

Wildcards can be used, such as U* for the job owner will match all users whose account starts with U. Based upon these criteria there are three useful policies that can be used for job interception:

No interception required

If there are no users in the SAP Systems that can release jobs, and no SAP applications are in use that create their own jobs, then interception serves no purpose and it does not need to be used.

Some jobs are intercepted

By carefully filling the criteria table, you can intercept only certain categories of jobs. For example, if you find that your financial end users are causing stoppages by ad-hoc selected jobs, you could put their usernames in the criteria table. Unfortunately the criteria do not easily allow you to match all background jobs started by SAP programs unless these have a job name that starts with a specific character string.

All jobs are intercepted

Usually the most useful policy is to intercept all jobs. When SAP CPS is used to do all batch job scheduling, the volume of jobs started outside of SAP CPS should be low. Every job started outside of SAP CPS has the potential of interfering with other jobs start time. To counter this, make sure every job is intercepted so that they will adhere to the SAP CPS queue and resource settings, ensuring fair sequencing of all batch jobs. This is also true for background jobs started by `DDIC`.

Parent/Child jobs

By default, even with BC-XBP 2.0, the CCMS does not track whether a job is submitted by a job or something else. This is unfortunate, because it means that job chains in SAP cannot say "start this when that job is finished completely, including its child jobs", since CCMS does not even know what a child job was.

With the new parent/child functionality one can tell CCMS to store the fact of the parent/child relationship of jobs. CCMS itself still does not offer any "wait for job plus children" functionality, so an external scheduler is the only way this can be done.

When the parent/child functionality is switched on, SAP CPS can retrieve the parent/child relationship. As with interception, it tells CCMS when it has made a permanent record of this relation so that the CCMS does not report the relation again.

Examples of batch jobs that are now easy to do include processing batch input folders.

Functionality completeness

With BC-XBP 2.0 it is now possible to set the job class, recipient and all print parameters per job step. You can also trigger batch input sessions by scheduling the ABAP program `RSBDCSUB` with the job definition *SAP_AbapRun*.

Availability of BC-XBP 2.0

The BC-XBP release that is available depends on both the SAP Basis (Web Application Server) release and the Support Package level. BC-XBP 2.0, the current release, is available for all SAP Web AS Basis releases that were actively supported by SAP in early 2004. It has been ported to all releases older than 6.30 by means of Support Packages (see SAP Note 604496).

SAP has fixed several issues relating to BC-XBP 2.0 in later Support Packages. Redwood therefore recommends using the Support Package of at least the Redwood recommended SP in the table below.

Release	Minimum SP	Redwood recommended SP
4.6B	SAPKB46B56	SAPKB46B56 and SAPKH46B60
4.6C	SAPKB46C41	SAPKB46C46

4.6D	SAPKB46D35	SAPKB46D35
6.10	SAPKB61027	SAPKB61038
6.20	SAPKB62017	SAPKB62033

BC-XBP 3.0

The latest release of the XBP interface adds much new functionality and improvements over XBP 2.0.

Note

Some SAP releases still report that they only support XBP up to 2.0, although they have XBP 3.0 as well. To make sure, execute `INITXBP2` in transaction **SE38**, you should see three options for job interception, if that is the case, you have XBP 3.0. You can force SAP CPS to use XBP 3.0 by specifying it in the SAP system object for job interception, event history and monitor synchronization.

Note

XBP 3.0 requires a logical system for the client, see Setting Up Logical Systems [http://help.sap.com/saphelp_nw70ehp1/helpdata/en/b4/b0b13bb3acef3ce10000000a11402f/frameset.htm] for more information on creating logical systems for clients.

Event History

XBP 3.0 introduces mapping of SAP events with external schedulers. SAP CPS allows you to map any event with a combination of SAP event and SAP event parameter. Again, wildcards can be used.

Spool Recipient

XBP 3.0 re-introduces the spool recipient which had been removed from the XBP 2.0 interface.

Variant Handling

XBP 3.0 introduces variant handling. You can now maintain variants from within SAP CPS, that is import, edit and save variants. You can also make use of temporary variants, which get deleted when the job finishes.

Interception

XBP 3.0 introduces a new criteria table. The new criteria table takes processing order into account and allows you to exclude certain jobs from interception. For example, you could intercept all jobs with a name starting with `B` except jobs starting with `Back`.

Availability of XBP 3.0

XBP 3.0 is available as of SAP Basis releases 7.0; it will be ported to a number of older releases via Support Packages. See SAP Note 961911 for more information.

Enhanced SAP Interfaces with Redwood Transports

Use

The Free of Charge version only supports native XBP functionality provided by the XBP release of the SAP System. This means that depending on the XBP release, it will have different XBP functionality. The transport files contain a number of functions that are stored in the Redwood name-space, thus they do not interfere with other functions in the SAP instance. They are used exclusively by SAP CPS.

SAP CPS takes advantage of transport files that are installed on the remote SAP System and that enhance the XBP and Industry Solutions (IS) functionality offered by that system.

The following illustrates the enhancements that transport files offer.

- Faster monitoring (all XBP releases)
- The standard SAP calls only allow SAP CPS to monitor jobs by doing a full table scan on the SAP job table. In larger systems, this can be quite time consuming. To improve performance, an optimized call is included that allows scanning for jobs changed since the previous scan.
- Extra print parameters (XBP 0.1, 1.0,2.0)
- Older XBP releases did not allow access to all job print parameters. This extends the functionality found in XBP 2.0 with two new print parameters.
- Dereleasing jobs (XBP 0.1, 1.0)
- Older XBP releases did not allow de-releasing jobs. This closely emulates the functionality found in XBP 2.0.
- External Alerting (XAL)

There was a problem in an XAL RFC call, where a table was not passed correctly. This made it impossible to receive alerts from SAP Systems. A corrected call is included in the transport file and is also available from SAP, see SAP Note 662694 for more information.

- Variant handling (XBP 0.1 - 3.0)

Variant handling has been improved with additional calls in the transport files. XBP 3.0 offers variant handling, but at the time of this writing, variant handling in SAP CPS is more reliable.

- Recipients (XBP 0.1 - 2.0)

It was not possible to get a list of recipients from an SAP System. Calls have been added to retrieve the list of recipients. Recipients retrieval functionality has been added to XBP 3.0.

When should a transport file be loaded

If one or more of the following conditions are applicable:

- You want to monitor other SAP jobs in SAP CPS, loading the XBP+ transport file is recommended.
- You want to intercept SAP jobs, and are running XBP 2.0, loading the XBP+ transport file is recommended.
- You want to intercept SAP jobs, and are running XBP 0.1 or 1.0, loading the XBP+ transport file is required.
- You want to receive CCMS alerts, loading the XAL+ transport file is required.
- You want to start and control mass activity runs on systems prior to ERP 2005S, loading the IS transport file is required.
- You want Variant handling, and you are running XBP 3.0 or lower, loading the XBP+ transport file is required.
- You want to work with Recipients, and are running XBP 2.0 or lower, loading the XBP+ transport file is required.

The XBP+ transport files

The XBP+ ("XBP Plus") transport file is delivered by Redwood to supply missing functionality in BC-XBP. Several different versions are available for different SAP releases:

SAP Basis/WAS releaseXBP+	Transport files
3.1i-4.0b	K000042.B31, R000042.B31
4.5b-4.6b	K900095.B45, R900095.B45
4.6c-710	K900099.I46, R900099.I46

The IS transport files

For Industry Solutions (IS) Mass activities, there is currently only one set of transport files `K900101.i46` and `R900101.i46` for releases *4.6.4* and *4.7.2*.

Note

These are the names of the transports at the time of this writing, the names change frequently as the transport files are updated.

Note

If you have a higher SAP basis version you must contact SAP Support Services for more information.

Procedure

Loading the transport files

1. Copy the K*.* file to */usr/sap/trans/cofiles* directory.
2. Copy the R*.* file to */usr/sap/trans/data* directory.
3. Open Transaction **STMS** in SAPGUI.
4. Choose *Import Overview* and choose the correct *Import queue*.
5. Add the request via menu item *Extra* > *Other Requests* > *Add*.
    a. The name of the request is the extension of the K*.* file followed by the name. For example, if the transport to be loaded consists of the files `R900099.I46` and `K900099.I46`, the transport request to specify is `I46K900090`.

Inspecting XBP Settings

1. Navigate to *Definitions* > *Job Definitions*.
2. Choose *Submit* from the context menu of the job definition *SAP_MaintainXbpOptions*.
3. Select the target *SAP System*, select *Show* for *Interception action* and *Parent/Child action*.
4. Choose *Submit* and inspect the output of the corresponding job.

Enabling XBP 2.0 Features

1. Navigate to *Definitions* > *Job Definitions*.

2. Choose *Submit* from the context menu of the job definition *SAP_MaintainXbpOptions*.
3. Select the target *SAP System*, select *Enable* for *Interception action* and *Parent/Child action*.
4. Choose *Submit* and inspect the output of the corresponding job, to verify if the settings were applied.

Enabling XBP 3.0 Features

1. Navigate to *Definitions* > *Job Definitions*.
2. Choose *Submit* from the context menu of the job definition *SAP_MaintainXbpOptions*.
3. Select the target *SAP System*, select *Enable with XBP 3.0 features* for *Interception action*.
4. Choose *Submit* and inspect the output of the corresponding job,to verify if the settings were applied.

SAP Communication Channels with PI/XI

Use

Communication channels are used in SAP instances to convert documents. These channels can be monitored from within the *Communication Channel Monitor*. SAP CPS uses SAP PI/XI interfaces to interact with communication channels.

Note

The SAP PI/XI interface requires a Chargeable license.

Communication channels can be controlled in three ways:

- Manual - You control the communication channel by choosing *Start* or *Stop* in the *Communication Channel Monitor*.
- Automatic - The communication channel is controlled by *availability times*.
- External - The communication channel is controlled by an external scheduler, such as SAP CPS.

This can be set in the *Communication Channel Monitor*.

You configure SAP CPS to control the channels as outlined in the Controlling Communication Channels with SAP CPS via PI/XI section of the documentation.

The following table lists the job definitions that are available to control communication channels:

Job Definition	**Description**
SAP_PI_GetChannelStatus	Retrieves the current status of an SAP XI/PI channel
SAP_PI_ImportChannels	Imports SAP XI/PI channels
SAP_PI_StartChannel	Starts an SAP XI/PI channel
SAP_PI_StopChannel	Stops an SAP XI/PI channel

Prerequisites

- You switched to external control in the Communication Channel Monitor [http://help.sap.com/saphelp_nw04/-helpdata/EN/44/2a1a8620323f0ee10000000a114a6b/content.htm].

- You configured SAP CPS to connect to the PI interface of your SAP instance, this is outlined in the Controlling Communication Channels with SAP CPS via PI/XI

Features

- You import communication channels by submitting the `SAP_PI_ImportChannels` job definition against the SAP instance. This job definition should be scheduled to run regularly, to make sure SAP CPS has an up-to-date list of communication channels.
- You check the status of communication channels by submitting the job definition `SAP_PI_GetChannelStatus` against the desired communication channel of the SAP instance.
- You can stop and start communication channels by submitting the job definitions `SAP_PI_StartChannel` and `SAP_PI_StopChannel`, respectively, against the desired communication channel of the SAP instance. These job definitions should be scheduled to allow automated control of the communication channels. If you have the Chargeable version and a free platform agent license, you can use file events to trigger the job. The communication channel is then started as soon as a specific file has been copied to a particular directory. This can be used to wait until all files have arrived before the communication channel is started, for example.

See Also

- Controlling Communication Channels with SAP CPS via PI/XI
- Importing Communication Channels

Configuring an SAP System for Central Process Scheduling

Use

To configure the Central Scheduler, you first need to have an account with Administrative privileges. Refer to the Creating Roles section for more information on creating a role.

When you connect to SAP CPS to monitor, intercept and submit jobs as well as other tasks, SAP CPS connects to the different SAP Systems with an RFC user to perform the requested tasks. So each SAP System that is to be managed by SAP CPS must have a user defined with sufficient authorizations. Depending on the required functionality, different SAP authorizations need to be granted to the RFC user. See the Reference section Required Authorizations of the Remote System User and the Assigning SAP Authorizations to the RFC User procedure for more information.

Note

You may be restricted to connecting to a single SAP System from within SAP CPS; this is the case if the value of the license key *ProcessServerService.SAP.limit* is `0` and your privileges.

Different interfaces are used to interact with the various SAP Systems in your landscape. XBP is one of them, to take full advantage of your SAP System, it is advised to enable XBP 2.0 available as of SAP Basis 4.6B with the latest Support Package from SAP.

- Different XBP releases explained
- Enable XBP 2.0 or 3.0 Functionality

Assigning SAP Authorizations to the RFC User

Use

The RFC user requires privileges on the satellite systems to schedule processes. These privileges are listed in the SAP Authorizations section in the Appendix of the following guide.

Procedure

1. Navigate to *Tools* > *Administration* > *User Maintenance* > *Role Administration* > *Roles* (transaction **PFCG**).
2. Create a new role *REDWOOD* and save the role.
3. Choose the Authorizations tab.
4. Choose *Change Authorization Data*. If the system shows a list of templates, choose *Do not select templates*.
5. You should now be in *Change role: Authorizations*.
6. Expand the main component to which the authorization belongs and locate the authorization.
7. Once you have located the authorization, choose the minus [-] to add it to your selection.
8. Choose *insert chosen* once you are done selecting authorizations.
9. Assign values to the properties of the authorization; you can save the role when the traffic lights are all green.
10. Choose *Save*.

Example

1. Navigate to *Tools* > *Administration* > *User Maintenance* > *Role Administration* > *Roles* (transaction **PFCG**).
2. Create a new role *MASALAN* and save the role.
3. Choose the Authorizations tab.
4. Choose *Change Authorization Data*. If the system shows a list of templates, choose *Do not select templates*.
5. You should now be in *Change role: Authorizations*.
6. Expand the *AAAB - Cross-application Authorization Objects* and locate *S_RFC - Authorization check for RFC access*.
7. Choose the minus [-] to add it to your selection.
8. Choose *insert chosen*.
9. Assign values to the properties of the authorization:
   a. Set *Activity* to *, you can choose the * to do this.
   b. Set *Name of RFC to be protected* to *.
   c. Set *Type of RFC object to be protected* to *Function group*.
10. Choose *Save*.

Enable XBP 2.0 or 3.0 Functionality

Use

To take full advantage of the capabilities of your SAP Basis release 4.6B or later with SAP CPS it is strongly advised to enable XBP 2.0 Parent/Child and Interception. An ABAP program named `INITXBP2` is available on SAP Basis release 4.6B with the latest Support Packages from SAP, later releases usually have these features enabled by default with the latest Support Packages from SAP. This program only needs to be executed once to enable XBP 2.0, which will allow you to enable parent-child-relationships for SAP jobs and SAP job interception. However, parent-child-relationships will not be visible in SAPGUI. Always use the job monitor in SAP CPS to see parent-child-relationships.

Note

If you have three options for job interception, then you have XBP 3.0. XBP 3.0 offers better job interception support among other improvements, see Understanding the Different XBP Interface Releases for more information.

You can also use the job definition *SAP_MaintainXbpOptions* to inspect and set XBP 2.0 or 3.0 settings.

Prerequisites

An SAP System with SAP Basis release 4.6B with the latest Support Packages.

Procedure

Setting and inspecting XBP settings with `INITXBP2`

1. Open transaction **SE38**
2. Fill `INITXBP2` into the program field
3. Choose *Execute* to run the program
4. Make sure *Interception* and *Parent-Child Function* are set to *on*
5. Choose *Execute* to save the settings.

Inspecting XBP Settings with *SAP_MaintainXbpOptions*

1. Navigate to *Definitions* > *Job Definitions*.
2. Choose *Submit* from the context menu of the job definition *SAP_MaintainXbpOptions*.
3. Select the target *SAP System*, select *Show* for *Interception action* and *Parent/Child action*.
4. Choose *Submit* and inspect the output of the corresponding job.

Enabling XBP 2.0 Features with *SAP_MaintainXbpOptions*

1. Navigate to *Definitions* > *Job Definitions*.
2. Choose *Submit* from the context menu of the job definition *SAP_MaintainXbpOptions*.
3. Select the target *SAP System*, select *Enable* for *Interception action* and *Parent/Child action*.
4. Choose *Submit* and inspect the output of the corresponding job, to verify if the settings were applied.

Enabling XBP 3.0 Features with *SAP_MaintainXbpOptions*

1. Navigate to *Definitions > Job Definitions*.
2. Choose *Submit* from the context menu of the job definition *SAP_MaintainXbpOptions*.
3. Select the target *SAP System*, select *Enable with XBP 3.0 features* for *Interception action*.
4. Choose *Submit* and inspect the output of the corresponding job,to verify if the settings were applied.

Connecting to SAP Systems

Use

SAP System definitions describe the characteristics of a single SAP System. It is possible to create more than one SAP System definition to connect to more than one SAP System. The SAP System definition is used to:

- Schedule/monitor SAP jobs via XBP, JXBP and BW-SCH interfaces.
- Synchronize monitoring and alerting data with the SAP System via XAL and XMW interfaces.
- Synchronize CCMS events, requires XBP 3.0.

Actions

SAP Systems support the following actions:

Action	**Description**
Export	Export the SAP System into a CAR file
Maintain Objects	Allows you maintain ABAP program variants, process chains, mass activities and set load-specific properties on the application servers
Edit	Edit the SAP System
Edit Security	Edit the security of the SAP System
Delete	Delete the SAP System
Duplicate	Make a copy of the SAP System to create a similar one
Expand All	Expand all SAP Systems in the current filter
New	Create a new SAP System
Filter > New Filter	Create a new SAP System filter
Filter > Edit Filter	Edit current SAP System filter
Filter > Delete	Delete current SAP System filter

Configuration

The configuration for each SAP System is managed through the *Environment > SAP Systems* view. No standard SAP Systems are provided with SAP CPS.

For each SAP System, you can configure the following:

- General settings like name and RFC connect string.
- One or more client connections for the XBP interface.

- Zero or more job control rules.
- Zero or one client connection for the XAL interface.
- Zero or one client connection for the XMW interface.
- Multiple Event Synchronization rules (requires XBP 3.0)
- One JXBP connection (in isolated mode, you cannot have two or more of XBP or JXBP connections in the same SAP CPS object; see the license restrictions below)

For most interfaces a username, password and client number are required. If you do not set an XBP account, SAP CPS will not be able to connect to the ABAP stack of the SAP instance.

Likewise, you need to set up accounts in the JAVA stack for JXBP (Java Scheduler connectivity).

Connect String

The connect string allows you to specify a subset of the settings you can specify in other SAP client tools:

- TYPE - (optional) The type of SAP system to connect to, only `R3` is supported.
- ASHOST - The application server host to connect to, this should be the message server.
- GROUP - (optional) Name of the SAP server group. This is a group of application servers in a load balancing connection. The default value is `PUBLIC`.
- GWHOST - (optional) The gateway host to use for the connection.
- MSHOST - The message server host of the SAP instance.
- R3NAME - The SID of the SAP instance.
- SYSNR - The system number of the SAP instance, it is usually the last two numbers of the port.
- TRACE - allows you to trace the RFC session, should be left on `0` unless SAP Support Services request you to change this value.

Note

Although it would be valid to specify a username and password as well as other settings, SAP CPS will discard them.

The following two combinations are frequently used:

Connection to a message server

```
MSHOST=<MS host> R3NAME=<SID> GROUP=<group>
```

Connection to a central or dialogue instance

```
ASHOST=<CI/DI-hostname> SYSNR=<Sys-Number> TRACE=0
```

You need several entries in your LOCAL /etc/services on the SAP CPS host for the remote SAP system for MSHOST connection strings.

- `sapms<SID> 36<instance id>/tcp`

- `sapdp<instance id> 32<instance id>/tcp`
- `sapgw<instance id> 33<instance id>/tcp`

Tip

For better performance, you should connect to the Message Server of the SAP instance.

Note

If you specify the *Application Server Host* in the connect string, you must not specify a *Message Server hostname* or the `R3NAME`.

When you create an SAP System definition, a process server and queue of the same name are automatically created with the required services, along with a queue provider that links the queue to the process server. When you delete an SAP System, the process server and queue are automatically removed. The process server and queue will share the same application when you create the SAP System, if you edit the SAP System later, you should change the application on the queue and scheduler manually, this allows you to easily export the SAP System via the application. Export requires the *Import/Export Module* which in turn requires the *Module.ImportExport* license key.

When you edit an SAP System, the required services are added automatically to the process server, so there is no need to change services on a process server.

The SAP configuration is read-only when the SAP process server starts up or is running, any changes in settings require a restart of the corresponding process server.

Note

By default, Mozilla Firefox browsers will offer to remember the password in the SAP System editor window. If you fill in two passwords on two tabs (XBP, XAL or XMW) in the SAP System editor, the Mozilla Firefox password manager assumes you want to change a password. You can either choose *Cancel* or disable *Remember password for sites* in the Mozilla Firefox Options/Preferences dialog, the latter is not recommended.

You may set a maximum setting for the number of RFC connections, please see the tuning for more information.

License Restrictions

If the license key *ProcessServerService.SAP.limit* license key is limited to `0`, SAP CPS is in isolated mode and you can only create one SAP System with one XBP client/username combination per isolation group. If this is the case, you must create an isolation group for the SAP System, assign the isolation group to yourself and log on with that isolation group prior to creating the SAP System inside SAP CPS.

Finding SAP Systems

You can search for SAP Systems using filters and the *Search SAP Systems* box on the SAP Systems tab. This box is known as the *intelliSearch* box and located under your username on the top right-hand side of the user interface. Filters allow you to specify a list of objects with static criteria. IntelliSearch allows you to specify complex queries in a simple way using prefixes. Prefixes are used to specify which property you are searching in and have short and long syntaxes. For example, if you want to display all SAP Systems with the term *import* in the comment, you would use the search criteria as follows:

```
c:import
```

You can search more than one property, as follows:

```
c:import n:Bi
```

Note

No spaces should be entered before or after the colon (:).

See the Advanced Object Search for more information.

The following table illustrates the available prefixes for SAP Systems:

Prefixes	Description
n, name	searches the name property
c, com, comment	searches the comment property
d, desc, description	searches the description property
a, app, application	searches the application property

Deleting SAP Systems

You can only delete an SAP System when no other objects relate to them. For example, if there are jobs that have run in the queue of the SAP System, the SAP System cannot be deleted until all jobs that ran in that queue have been deleted. You can see all jobs that relate to the queue in *Related Objects* in the lower detail pane and on the show page of the queue. Besides, You cannot delete an SAP System if the process server is still running, please stop the process server before you delete the SAP System.

The table in related objects contains three columns:

- Type - the type of object with a link to it
- Related Object - the name of the object with a link to it
- Used As - objects can sometimes be used in different roles

Security

Privilege	Description
SAPSystem.Create	Create SAP Systems
SAPSystem.Delete	Delete SAP Systems
SAPSystem.Edit	Edit SAP Systems
SAPSystem.View	Access SAP Systems
SAPSystem.BusinessAutomationAPI	Use the SAP Applications API

You can grant privileges on two levels, *Access* and *Admin;* a privilege granted on *Admin* level allows the grantee to grant the privilege to other users. These privileges can be granted *system-wide*, per *partition* or *isolation group*.

If you have the security module, which requires the *Module.Security* license key, you have an additional *Security* tab on the SAP System. It allows you to specify which users can access, edit, and delete the SAP System.

- Granting and Revoking System Privileges
- Granting or Revoking Object Privileges

Prerequisites

- A user with sufficient privileges in the Central scheduler `http://<server>:<port>/scheduler`.
- The SAP R/3 connect string to the SAP System.
- Client / username and password of the RFC user of the desired interfaces.
- XBP versions supported in the SAP System (XBP 0.1, 1.0, 2.0, 3.0) (for parent child dependencies the report `INITXBP2` must be run).
- The SAP System must have a default client configured, please refer to the SAP documentation.
- A license for the additional SAP process server.
- An available shared partition if the license key *System.sharedPartitions* is set to a numeric value.

Procedure

To create a connection for starting jobs from SAP CPS only, without job interception, event handling, or Monitor synchronization, perform as follows:

1. Choose *Environment* > *SAP Systems* in the Navigation bar.
2. Choose *New* from the context menu.
3. On the *SAP System* tab, enter a *Name*, *Connect String*, and a *Time zone*,
4. On the *XBP* tab, fill in *Client Number*, *Username*, *password*, and check *Default Client*
5. Choose *Save & Close* to save the changes and close the edit window.

To start ABAP programs from SAP CPS in an SAP System, refer to the following procedure:

Submitting a job or job chain

To trigger batch input sessions in an SAP System, refer to the following procedure:

Controlling Batch Input Sessions

To monitor or intercept (take control) of jobs in an SAP System, refer to the following procedure:

Intercepting SAP Jobs

To map SAP CPS events to events in an SAP System, refer to the following procedure:

Interacting with SAP Events

To map return codes to specific job log entries, refer to the following procedure:

Parsing Log Files to set Job Status of SAP Jobs

To import one or more monitors from an SAP System, refer to the following procedure:

Integrating SAP Monitors

To send SAP CPS monitoring data to an SAP System, refer to the following procedure:

Sending Monitoring Data to SAP Systems

To control communication channels in an SAP System, refer to the following procedure:

Controlling Communication Channels with SAP CPS via PI/XI

To monitor SAP CPS availability from the CEN refer to this procedure:

Monitoring SAP CPS Availability

Note

When you create or edit an SAP System definition, you have to restart the corresponding process server.

For every SAP System, at least one default client must exist.

Every SAP System created in SAP CPS requires the settings mentioned in Connection Details for Connecting to the ABAP Stack, even if you only want to synchronize monitors!

Values

For information on the necessary fields of the tabs that you need to fill-in in order to be able to connect to the ABAP stack of an SAP instance, see the Connection Details for Connecting to the ABAP Stack section of the documentation.

Example

You need to create a connection to an SAP System running SAP BW 6.40.

You have the following information:

- The SAP R/3 connect string is: `MSHOST=169.254.10.12 R3NAME=PR1 GROUP=SPACE`
- The Client is 200, username is masalan and password is masalan
- XBP version supported is XBP 2.0 (INITXBP2 has run).
- The default client of the SAP System is client 200.

1. Choose *Environment > SAP Systems* in the Navigation bar.
2. Choose the *New* button and enter the information below (listed per tab)
3. Choose *Environment > Process Servers*
4. From the context menu of the process server that belongs to the newly create SAP System connection, choose Start.

Tab	**Field**	**Value**
SAP System	Name	sapbw_prd
SAP System	Connect String	`MSHOST=169.254.10.12 R3NAME=PR1 GROUP=SPACE`
SAP System	Job Handling	All
SAP System	Time zone	Europe/Berlin
SAP System	Application	SAPBW

SAP System	Description	Production SAP BW system in Berlin
XBP	Client Number	200
XBP	Username	masalan
XBP	Password	******* (masalan in this example)
XBP	Audit level	0 - Change actions
XBP	SAP Language	EN
XBP	Description	Client for the XBP interface
XBP	Default Client	selected
XBP	Max RFC Connections	

Installing the JCO Libraries

Use

SAP System definitions describe the characteristics of a single SAP System. It is possible to create more than one SAP System definition to connect to more than one SAP System. However connection to an SAP system requires the installation of the JCO library. This library has not been found, or an out of date version has been found. SAP JCO version 2.1.8 or newer is required.

Installation Instructions

1. Navigate to http://service.sap.com/connectors; you will need your SAP Service Marketplace login (for example SAP S-User) to log in.

2. Choose `SAP Java Connector`

3. In the tree on the left-hand side, choose `Tools & Services`

4. Download the appropriate binary for the system, for example: `Windows NT / Windows 2000 / Windows 2003: 32bit SAP JCo 2.1.x` zip file.

For a Redwood Platform installation on Windows NT / Windows 2000 / Windows 2003: 32bit SAP JCo 2.1.x follow these steps:

1. Copy the downloaded file (`sapjco-ntintel-2.1.x.zip`) to:

```
C:\Program Files\Redwood\Server\j2ee\cluster\global\tomcat\sapjco-ntintel-2.1.8.zip
```

1. Restart the Scheduler

For other systems ensure that the `sapjco.jar` is on the Java classpath (e.g. in the Application Server library path) and that the `sapjcorfc` (for example `sapjcorfc.dll`) and `librfc` (for example `librfc32.dll`) libraries are on the linkable path. You can put these into the `jre/bin` directory.

Note

If you encounter dependency problems, see SAP Note 684106 on Windows. On UNIX/GNU/Linux, use `ldd` (GNU/Linux), `otool` (Mac OS X), or the native dependency checker for your platform.

Connection Details for the ABAP Stack

Use

The Following tabs of the SAP System editor must contain connection-related information if you want to connect to the ABAP stack of an SAP instance.

Setting up connection information

Tab	Field	Description
Partition	Partition of the SAP system	
SAP System	Name	The name for the SAP System. This name is also used to create the corresponding SAP process server and queue. For instance, if the name of the SAP System definition is *TST*, then the created process server gets the name *TST_ProcessServer* and the queue is named *TST_Queue*.
SAP System	App, server load balancing	When this is enabled, SAP CPS will balance the load across application servers.
SAP System	RFC Connect String	RFC connection definition used to connect to the SAP System via SAP Java Connector (JCo).
SAP System	J2EE URL	The URL used by the JXBP and PI/XI interfaces for connecting to the JAVA stack of SAP NetWeaver
SAP System	Job Handling	Job handling allows you to specify which jobs the job control rules should apply to. If you set it to *Intercepted*, only jobs intercepted by SAP will be taken into account. If you set it to *All*, SAP CPS will attempt to apply the rules on all jobs in the SAP System. Note that if you want to just monitor jobs, this setting should be set to *All*.
SAP System	Default Job Time Zone	This field is used for detecting the time zone of the SAP instance. The default job time zone for jobs submitted in SAP CPS is the time zone of the original request time.
SAP System	Application	The name of the application used to group this object.
SAP System	Description	The description of the SAP System.
Comment	Comment	A comment about the SAP System.

Time zone data

The time zone of the SAP system is detected via the offset. If the current offset matches the current offset of the time zone specified in *Default Job Time Zone*, then SAP CPS uses that time zone when interpreting dates from SAP instances (used for *Remote Start Time*, for example). If the two do not match, SAP CPS uses the current offset of the SAP instance as the time zone, like *GMT+,<offset>*.

Setting up client/username combinations

At least one client/username combination is required and you must specify one combination as being the default.

Tab	Field	Description
XBP	Client number	SAP client number as a three digit number.

XBP	Username	The RFC user name. This user needs certain authorizations in the SAP System, as described below in the Reference section under Required Authorizations of the Remote System User. Of course it is possible to use here the same user as for XAL and XMW. In this case the user needs all authorizations required by these interfaces.
XBP	Password	Password of the RFC user.
XBP	Check Connection Settings	Allows you to check that the *RFC Connect String*, *Client Number*, *Username* and *Password* are correct.
XBP	Enabled	Allows you to disable the client, keeping the settings for future use
XBP	Default client	Specify if this client is the *default client*. The default client is used by the SAP process server for the standard connection to the SAP System to start/monitor/intercept jobs. Please make sure that only one client is marked as default. If the default client check box is selected in multiple XBP interfaces, the one with the lowest client number is used by default when required. Yet to avoid confusion it is better to make sure that only one client is designated as default client.
XBP	Audit level	The XBP audit level (the audit log is visible via SAP transaction **RZ15**) of the interface, and is an integer with the value range of 0-3, with higher numbers denoting more verbose audit level; only to be used when SAP Support instructs you to do so.
XBP	SAP language	The language that the SAP process server uses to connect to the SAP System. This is also the default step language for ABAP jobs, if the corresponding parameter in the job definition is left empty.
XBP	Description	the description to use for this rule. If the description is not set, the rules are named <Client number>/<Username>.
XBP	Max. RFC connections	Specifies how many RFC connections the SAP process server opens for this client. If omitted, the number of connections are not limited. The SAP process server uses internally a connection pool, so the open connections are reused, and also closed if they are idle for a certain amount of time.
XBP	Interface version	The version of the XBP interface that the SAP process server should use. If this field is left on *automatic*, the highest available interface version is used. Note that older 7.0 SAP releases will report the highest version as being XBP 2.0 although XBP 3.0 is available. It is highly recommended to upgrade to the latest Support Package, or at the least Support Package 14.

Connect to an SAP System with SNC

Use

Secure Network Communications (SNC) can be used to secure RFC connections between SAP CPS and SAP instances.

To make use of SNC, you need to configure your SAP system to use SNC.

This topic will discuss the configuration instructions for the ADS Kerberos infrastructure on Microsoft Windows,

please see the relevant SAP documentation for your platform/infrastructure.

The systems involved require the following Kerberos dll to be available in the path.

- gsskrb5.dll for 32bit.
- gx64krb5.dll for x64.

These files can be found on ftp://ftp.sap.com/pub/ietf-work/gssapi.

Prerequisites

- You have installed the correct Kerberos DLL for your platform in the path of your systems.

Procedure

1. In the instance profile of the central instance (Target SAP System), the SAP parameters below need to be set.
2. Restart the SAP instance to activate the SNC settings.

```
snc/enable = 1
snc/gssapi_lib =<DRIVE>:\%windir%\system32\gsskrb5.dll
snc/identity/as =p:<SAP_Service_User>@<DOMAIN_NAME>
```

Configuration on the ABAP System

1. In the Target ABAP System, define a service user that will be used by SAP CPS to connect using transaction **SU01**.
2. On the SNC-Tab, specify the SNC Name for SAP CPS (p:<SAP_Service_User>@<DOMAIN_NAME>).

Note

If more than one SNC name needs to be assigned to this user, these can be specified in table `USRACLEXT` (use transaction **SM30** to maintain it).

Configuration SAP CPS:

1. Navigate to *Environment > SAP Systems*
2. Choose *Edit* from the context menu of the SAP system you want to connect to via SNC.
3. On the *SAP System* tab, fill the connect string below into the *RFC Connect String* field.
4. Choose *Check connection* and make sure the connection was successful.
5. On the *XBP* tab, choose *Check Connection Settings*.
6. Choose *Save & Close*.
7. Navigate to *Environment > Process Servers* and restart the process server belonging to the SAP system.

```
SNC_MODE=1 SNC_PARTNERNAME="p:<SAP_Service_User>@<DOMAIN_NAME>" SNC_LIB=<KERBEROS_DLL>
```

Example

This example shows how a Microsoft ADS Kerberos infrastructure can be used to provide Single-Sign-on between SAP CPS and a connected SAP ABAP System.

Instance profile settings in SAP

```
snc/enable = 1
snc/gssapi_lib =c:\windows\system32\gsskrb5.dll
```

```
snc/identity/as =p:SAPServicePRD@MASALAN.COM
```

Service user in the target ABAP system

```
p:SAPServicePRD@MASALAN.COM
```

Connect String in SAP CPS

```
SNC_MODE=1 SNC_PARTNERNAME="p:SAPServicePRD@MASALAN.COM" SNC_LIB=gsskrb5.dll
```

Intercepting SAP Jobs

Use

Often jobs are scheduled by users and interfere with other processes or need to follow complex rules prior to execution. Job interception allows SAP CPS to take control of the job and to let it run based on previously defined criteria. Job interception can be done in two ways:

- by the SAP System itself.
- by SAP CPS, also called late intercept.

In the first instance, the SAP System compares the job to a list of criteria and decides to execute the job or to "derelease" the job, the latter means that the job will not run at this point. SAP CPS will take control of "dereleased" jobs and reschedule them. This functionality requires XBP 2.0 or later. With XBP 3.0 your interception rules use the processing order, so you can easily include and exclude certain jobs from interception.

Note

XBP 3.0 requires a logical system for the client, see Setting Up Logical Systems [http://help.sap.com/saphelp_nw70ehp1/helpdata/en/b4/b0b13bb3acef3ce10000000a11402f/frameset.htm] for more information on creating logical systems for clients.

In the second instance, SAP CPS will attempt to intercept the job before SAP has a chance to execute it. This is not always reliable, this functionality requires that the Redwood transport files be loaded.

You should always synchronise intercept criteria with the SAP system when you have changed it.

Note

If your SAP system supports XBP 3.0, you should decide which tool of SAP CPS or SAP `criteria_manager` to use for defining intercept criteria and exclusively use that.

Job Handling Field

The job handling field on an SAP system allows you to apply the job control rules to jobs intercepted by SAP, all jobs or none (disbaled). When job handling is disabled, your job control rules will be ignored and no jobs will be picked up on the SAP side.

Connect String

The connect string allows you to specify a subset of the settings you can specify in other SAP client tools:

- TYPE - (optional) The type of SAP system to connect to, only `R3` issupported.

- ASHOST - The application server host to conect to, this should be the message server.
- GROUP - (optinal) Name of the SAP server group. This is a group of application servers in a load balancing connection. The default value is `PUBLIC`.
- GWHOST - (optional) The gateway host to use for the connection.
- MSHOST - The message server host of the SAP instance.
- R3NAME - The SID of the SAP instance.
- SYSNR - The system number of the SAP instance, it is usually the last two numbers of the port.
- TRACE - allows you to trace the RFC session, should be left on `0` unless SAP Support Services request you to change this value.

Note

Although it would be valid to specify a username and password as well as other settings, SAP CPS will discard them.

The following two combinations are frequently used:

Connection to a message server

```
MSHOST=<MS host> R3NAME=<SID> GROUP=<group>
```

Connection to a central or dialogue instance

```
ASHOST=<CI/DI-hostname> SYSNR=<Sys-Number> TRACE=0
```

You need several entries in your LOCAL /etc/services on the SAP CPS host for the remote SAP system for MSHOST connection strings.

- `sapms<SID> 36<instance id>/tcp`
- `sapdp<instance id> 32<instance id>/tcp`
- `sapgw<instance id> 33<instance id>/tcp`

Note

If you specify the *Application Server Host* in the connect string, you must not specify a *Message Server hostname* or the `R3NAME`.

Tip

For better performance, you should connect to the Message Server of the SAP instance.

Prerequisites

- Client / username and password of the RFC user.
- XBP versions supported in the SAP System (XBP 0.1, 1.0, 2.0, 3.0) (for parent child dependencies the report `INITXBP2` must be run).
- The SAP System must have a default client configured, please refer to the SAP documentation.

Procedure

To monitor jobs in an SAP System:

1. Choose *Environment > SAP Systems* in the Navigation bar.
2. Choose the SAP System name in the overview table
3. Choose *Edit* from the context menu
4. On the *SAP System* tab, enter a *Connect String*, a *Time zone*, and set *Job Handling* to *All*
5. On the *XBP* tab, fill in *Client Number*, *Username*, *password*, and check *Default Client*, if no default client has been previously specified.
6. On the *Job Control Rules* tab, choose *Monitor* for *Action*, fill in a valid client mask and username
7. Choose *Save & Close* to save the changes and close the edit window.

To intercept jobs in an SAP System:

1. Choose *Environment > SAP Systems* in the Navigation bar.
2. Choose the SAP System name in the overview table.
3. Choose *Edit* from the context menu.
4. On the *SAP System* tab, enter a *Connect String*, a *Time zone*, and set *Job Handling* to *All*.
5. On the *XBP* tab, fill in *Client Number*, *Username*, *Password* and check *Default Client*, if no default client has been previously specified.
6. On the *Job Control Rules* tab, choose *Schedule* for *Action*, fill in a valid client mask and username.

   a. With XBP 2.0 enabled, all jobs matching the client and username masks will be intercepted in the Central Scheduler.

7. Choose *Save & Close* to save the changes and close the edit window.

Synchronize Intercept Criteria with the SAP System:

1. Choose *Definitions > Job Definitions* in the Navigation bar.
2. Choose the job definition named *SAP_SynchronizeInterceptionCriteria* in the overview table.
3. Choose *Submit* from the context menu, set the target SAP system, select a direction and choose *Submit*.

Note

You should always choose *Append* unless you are sure of what you are doing. You will overwrite interception data in the SAP system.

Tip

Run the job definition *SAP_SynchronizeInterceptionCriteria* with the action *Show only* to inspect the current settings and make sure you are not deleting data you need.

Values

Configuration information is available in the Job Interception Configuration section of the documentation.

Example

You need to edit an SAP System running against SAP BW 6.40 to intercept jobs.

You have the following information:

- The SAP R/3 connect string is: `MSHOST=169.254.10.12 R3NAME=PR1 GROUP=SPACE`
- The Client is 200, username is masalan and password is masalan
- XBP version supported is XBP 2.0.
- The default client of the SAP System is client 200.

1. Choose *Environment* > *SAP Systems* in the Navigation bar.
2. Choose the *Edit* button and enter the information below
3. Choose *Environment* > *Process Servers*
4. From the context menu of the process server that belongs to the newly create SAP System connection, choose Start.

Tab	**Field**	**Example**
SAP System	Name	sapbw_prd
SAP System	Connect String	`MSHOST=169.254.10.12 R3NAME=PR1 GROUP=SPACE`
SAP System	Job Handling	All
SAP System	Time zone	Europe/Berlin
SAP System	Application	SAPBW
SAP System	Description	Production SAP BW system in Berlin
XBP	Client Number	200
XBP	Username	masalan
XBP	Password	******* (masalan in this example)
XBP	Audit level	0 - Change actions
XBP	SAP Language	EN
XBP	Description	Client for the XBP interface
XBP	Default Client	selected
XBP	Max RFC Connections	
XBP	Interface Version	2.0
XBP Job Control Rules	Processing order	0
XBP Job Control Rules	Action	Schedule
XBP Job Control Rules	Client Number Mask	20*
XBP Job Control Rules	Job Name Mask	RSUSR*
XBP Job Control Rules	Username Mask	*

XBP Job Control Rules	Description	Intercept jobs starting with RSUSR in their name
XBP Job Control Rules	Default Queue	
XBP Job Control Rules	Priority	50

Job Interception Configuration

Use

When a job is found in the SAP System it is checked against the XBP interception rules defining the action to take on the job.

Setting up connection information

See the Connection Details for the ABAP Stack section of the documentation for details on how to set up a connection to the ABAP stack.

Setting up the job handling information

Tab	Field	Description
XBP Job Control Rules	Job Handling	Reports the setting of the *Job Handling* field on the SAP System tab.
XBP Job Control Rules	Enabled	Allows you to disable rules, preserving the settings for future use.
XBP Job Control Rules	Description	the description to use for this rule. If the description is not set, the rules are named <Action>/<Client Number Mask>/<Username Mask>/<Job Name Mask>, Please see the list below for available actions.
XBP Job Control Rules	Processing order	Order in which to process this rule.
XBP Job Control Rules	Action	Defines how to handle the job. Please see the list below for available actions.
XBP Job Control Rules	Client Number Mask	SAP client pattern that is used to match jobs.
XBP Job Control Rules	Job Name Mask	Job name pattern that is used to match jobs.
XBP Job Control Rules	Username Mask	User name pattern that is used to match jobs.
XBP Job Control Rules	Job class (A, B, C)	The SAP job class
XBP Job Control Rules	Default Queue	Queue the job should be put on (currently not used).
XBP Job Control Rules	Priority	Priority of the job (currently not used).
XBP Job Control Rules	Retrieve Job log	Should the log of the job be retrieved?
XBP Job Control Rules	Retrieve Spool	Should the output file of the job be retrieved?
XBP Job Control Rules	Applies to jobs in SAP status	You may restrict this rule to jobs with a specific status in SAP, see the details below.

The following wildcards are supported in masks:

- *** to match any string, for example `BI_PROCESS_*` would match `BI_PROCESS_ABAP` and `BI_PROCESS_OR`.
- *?* to match a single character, for example `RSUSR?00` would match `RSUSR200` and `RSUSR400`.

The following actions are available:

Action	Description
Monitor	A reference to the job is created in SAP CPS. The status of the SAP job can be monitored in the job monitor of SAP CPS.
Hold	A reference to the job is created in SAP CPS. If the SAP job is found to be in a *Released* state in SAP (which means waiting to start at a certain point in time), it is dereleased (put back in status *Scheduled* in SAP) and submitted on hold in SAP CPS.
Ignore	Jobs that match this rule are *confirmed* in SAP, so they are not checked again, and ignored (no reference to the job is created in SAP CPS).
Schedule	A reference to the job is created in SAP CPS. If the SAP job is found to be in a *Released* state in SAP (which means waiting to start at a certain point in time), it is dereleased (put back in status *Scheduled* in SAP). The job is scheduled to start in SAP CPS at the time it was intended to be in SAP.
Start Immediately	A reference to the job is created in SAP CPS. If the SAP job is found to be in a *Released* state in SAP (which means waiting to start at a certain point in time), it is dereleased (put back in status *Scheduled* in SAP) and SAP CPS attempts to start the job as soon as possible.
Schedule Intercepted	Jobs that have been intercepted by the SAP system are scheduled in SAP CPS.

Note

Jobs that do not match any of the defined rules are skipped, but not *confirmed* in SAP, so they are checked at the next synchronization round.

You can restrict job control rules to SAP jobs with the following statuses:

- Scheduled (P)
- Released (S)
- Ready (Y)
- Active (R)
- Finished (F)
- Canceled / Error (A)

Tip

You can check all job control rules that use a specific default queue across all SAP systems in the *Related Objects* list of the queue in question.

Controlling Batch Input Sessions

Use

Batch Input Sessions are a special type of SAP job that process incoming data. A particular OS directory is designated in SAP as the Batch Input Folder. Other applications can then send data in a special SAP format called *iDocs* to the SAP System by placing a file in this directory.

To process the incoming data, you submit the ABAP program `RSBDCSUB`. This program reads the batch input folder and submits a new SAP job for every file found in this folder. Unfortunately, `RSBDCSUB` does not wait for the processing jobs to finish.

It is, therefore, very difficult to build dependencies within SAP using `RSBDCSUB`, because the start job does not wait for the processing to finish and it is not known beforehand what the child jobs will be. With the standard CCMS scheduler you are forced to resort to ABAP programming to find the child jobs and then wait for them to finish.

If you use BC-XBP 2.0, SAP CPS will *automatically* realize that `RSBDCSUB` has created child jobs and will automatically delay setting the *RSBDCSUB* job to a finished state until all these jobs are finished. So when BC-XBP 2.0 is used, you can schedule `RSBDCSUB` just like any other job. This behavior also works when the *RSBDCSUB* job is intercepted. In both cases any dependencies made on this job in SAP CPS will only start once all child jobs are finished.

Procedure

1. Navigate to *Definitions* > *Job Definitions*.
2. Choose *Submit* form the context menu of the *SAP_AbapRun* job definition.
3. Fill in *SAP System, SAP job name* and enter `RSBDCSUB` into the *ABAP program* field.
4. Choose *Submit* and navigate to *Monitoring* > *Jobs* to locate the jobs.

Interacting with SAP Events

Use

XBP 3.0 introduced new functionality known as *Event History* and which allows SAP CPS to interact with *SAP Batch Events*. SAP CPS events can now be mapped to *SAP Batch Events*. The state of SAP CPS events is constantly synchronized with an *SAP Batch Events* and parameter-combinations. Furthermore, *Event History* also keeps track of events, logging these for reference; this means that it is now possible to see if an event was raised in the past and when. You can define which events are used for event history in your criteria profile in SAP as well as define the events in SAP CPS. See the Criteria Manager [http://help.sap.com/saphelp_nw70/helpdata/en/44/bca798e3d756a4e10000000a1553f6/content.htm] section of the SAP Help for more information on defining event synchronization in SAP.

You define the events to be synchronized on the *XBP Event Handling* tab of the SAP system object. Once you defined your event matching, you synchronize the settings with the SAP instance using the job definition SAP_SynchronizeEventHistoryCriteria.

To enable *Event History* on a specific SAP instance, the instance must have XBP 3.0 or higher. Please consult your SAP documentation to find out if your instance qualifies.

The *XBP Event Handling* tab requires an RFC user with sufficient privileges as outlined in the Reference SAP RFC User Privileges.

Note

XBP 3.0 requires a logical system for the client, see the Setting Up Logical Systems [http://help.sap.com/saphelp_nw70ehp1/helpdata/en/b4/b0b13bb3acef3ce10000000a11402f/frameset.htm] section for more information on creating logical systems for clients.

Prerequisites

- An SAP System with XBP 3.0 enabled
- RFC User credentials with sufficient privileges

Procedure

1. Navigate to *Environment* > *SAP Systems*.
2. Choose *Edit* from the context menu of the SAP System.
3. Choose the *XBP Event Handling* tab and fill in the details.
4. Choose *Save & Close*
5. Navigate to *Environment* > *Process Servers*.
6. Choose *Stop* then *Start* from the context menu of the corresponding process server. Please wait for the process server to stop completely before starting it again.

Synchronizing event-handling rules with the SAP system

1. Navigate to *Definitions* > *Job Definitions*.
2. Choose *Submit* from the context menu of *SAP_SynchronizeEventHistoryCriteria*.
3. Fill in the parameters.
4. Choose *Submit*.

Values

Information on the required settings for setting up event synchronization can be found in the Configuration Details on Interacting with SAP Events section of the documentation.

See Also

Criteria Manager [http://help.sap.com/saphelp_nw70/helpdata/en/44/bca798e3d756a4e10000000a1553f6/-content.htm]

Configuration Details on Interacting with SAP Events

Use

The following configuration data needs to be used to interact with SAP events via the XBP 3.0 interface.

XBP event control rules control which events are raised in SAP CPS when an event is raised in SAP.

Setting up connection information

Tab	**Field**	**Description**
SAP System	Name	The name for the SAP System. This name is also used to create the corresponding SAP process server and queue. For instance, if the name of the SAP System definition is *TST*, then the created process server gets the name *TST_ProcessServer* and the queue is named *TST_Queue*.
SAP System	Connect String	RFC connection definition used to connect to the SAP System via SAP Java Connector (JCo).
SAP System	Job Handling Enabled	Is job handling enabled for this SAP System, if enabled the intercepted or all jobs are shown in the job monitor.

SAP System	Default Job Time Zone	This field is used for detecting the time zone of the SAP instance. The default job time zone for jobs submitted in SAP CPS is the time zone of the original request time.
SAP System	Application	The name of the application used to group this object.
SAP System	Description	The description of the SAP System.
Comment	Comment	A comment about the SAP System.

Time zone data

The time zone of the SAP system is detected via the offset. If the current offset matches the current offset of the time zone specified in *Default Job Time Zone*, then SAP CPS uses that time zone when interpreting dates from SAP instances (used for *Remote Start Time*, for example). If the two do not match, SAP CPS uses the current offset of the SAP instance as the time zone, like *GMT+,<offset>*.

Setting up client/username combinations

At least one is required, one client/username combination must be configured as the default.

Tab	Field	Description
XBP	Client number	SAP client number as a three digit number.
XBP	Username	The RFC user name. This user needs certain authorizations in the SAP System, as described below in the Reference section under Required Authorizations of the Remote System User. Of course it is possible to use here the same user as for XAL and XMW. In this case the user needs all authorizations required by these interfaces.
XBP	Password	Password of the RFC user.
XBP	Audit level	The XBP audit level (the audit log is visible via SAP transaction **RZ15**) of the interface, and is an integer with the value range of 0-3, with higher numbers denoting more verbose audit level; only to be used when SAP Support instructs you to do so.
XBP	SAP language	The language that the SAP process server uses to connect to the SAP System. This is also the default step language for ABAP jobs, if the corresponding parameter in the job definition is left empty.
XBP	Description	the description to use for this rule. If the description is not set, the rules are named <Client number>/<Username>.
XBP	Default client	Specify if this client is the *default client*. The default client is used by the SAP process server for the standard connection to the SAP System to start/monitor/intercept jobs. Please make sure that only one client is marked as default. If the default client check box is selected in multiple XBP interfaces, the one with the lowest client number is used by default when required. Yet to avoid confusion it is better to make sure that only one client is designated as default client.
XBP	Max. RFC connections	Specifies how many RFC connections the SAP process server opens for this client. If omitted, the number of connections are not limited. The SAP process server uses internally a connection pool, so the open connections are reused, and also closed if they are idle for a certain amount of time.
XBP	Interface version	The version of the XBP interface that the SAP process server should use. If this field is left empty, the highest available interface version is used.

Mapping SAP CPS Events to SAP Events

Tab	Field	Description
XBP Event Control Rules	SAP Event Id	The SAP Event Id
XBP Event Control Rules	SAP Event Parameter	The parameter of the SAP event
XBP Event Control Rules	Event to Raise	The event to raise in SAP CPS
XBP Event Control Rules	Description	The description of the rule. If this is not set the description is <SAP Event Id>/<SAP Event Parameter>/<Event to Raise>

Parsing Log Files to set the Job Status of SAP Jobs

Use

SAP jobs reach status *Completed* when no error was returned and *Error* when an error occurred. You can force an SAP job into *Error* or *Completed* based on a specific message in the log file. This is done by mapping specific messages to return codes; the return code `0` maps to *Completed*, other return codes map to *Error*. The return code can be configured per SAP instance, on the *Return Code Mapping* tab of SAP Systems in SAP CPS.

Matching a Log Entry

The following information is required to match a log entry:

- Message Class - The type of message; it can be found in the log, usually the third column.
- Message Number - The number of the message; it can be found in the log, usually the fourth column.
- Message Type - The type of the message, see list below. The message class and number usually uniquely identify the message, so this can be set to asterisk (*) in most cases.
- Message Text (optional) - The text of the message, it can be found in the log.

The message type can be any of the following:

- A - Termination
- E - Error Message
- I - Information
- S - Status Message
- W - Warning
- X - Exit

The message type can only be retrieved if one of the following is true:

- transports are used
- if the SAP Note 603919 - *Reading job log by XBP: Message type is missing* is implemented
- 6.40 or newer SAP releases

Based on the criteria, SAP CPS parses the log file and when it finds a matching entry, it will set the return code of the job to the one configured for the rule. Optionally, you may have SAP CPS create a job note, which will inform operators, that return code mapping was used on this job. All criteria must be fulfilled for the match to be successful.

The following is the contents of a log file:

```
180006 20100613 00 516 Job started  0
180007 20100613 00 550 Step 001 started (program BTCTEST, variant , user ID REDWOOD)  0 001 BTCTEST  REDWOOD
180007 20100613 00 544 TEST: SY-BATCH indicates background processing  0
180007 20100613 00 517 Job finished  0
```

In the above message, we see the following message `180007 20100613 00 544 TEST: SY-BATCH indicates background processing 0` that indicates that no background processing jobs were created, so we would like to force the SAP job into error.

The following rules would catch this log entry:

- Message Class - `00`
- Message Number - `544`
- Message Type - `S`
- Message Text - `TEST: SY-BATCH*`

It is easier with SAP error messages, see the following error:

```
BAPI exception: RFC function module = BAPI_XBP_NEW_FUNC_CHECK, message type=E, id=XM,
number=233, text=I:XM:233 BAPI_XBP_NEW_FUNC_CHECK,
msgv1=BAPI_XBP_NEW_FUNC_CHECK, msgv2=, msgv3=, msgv4=
```

The following rule would match this log entry:

- Message Class - `XM`
- Message Number - `233`
- Message Type - `E`
- Message Text - `I:XM:233 BAPI_XBP_NEW_FUNC_CHECK*`

Specifying the Jobs to be Parsed

The following criteria is available to specify which job logs are to be parsed:

- Client Number Mask - a client number mask can be a specific client or a range of clients defined with wildcards. For example, `2*` would match clients `200` and `250`, but not `300`.
- Jobname Mask - a job name mask can be a specific job name, or range of job names specified with wildcards. For example, `BI *` would match `BI PostProcessing` but not `Backup Production System 12/03`.
- Username Mask - a username mask should be left as `*` unless you are sure the job will always be started by the same user.

This allows you to limit the rule to a more or less specific range of jobs. For performance reasons, SAP recommends to avoid parsing logs of all jobs. All criteria must be fulfilled for the match to be successful.

When you configure more than one rule, you should make sure that the most restrictive rules need to have a lower processing order than more generic rules.

Procedure

1. Navigate to *Environment* > *SAP Systems*.
2. Choose '"Edit" from the context menu of an SAP System.
3. Choose the *Return code mapping* tab.
4. Choose *New* and fill in the details of the rule, check the *Values* section below for a description of the available fields.
5. Choose *Save & Close* when you are done.

Values

Field	**Description**
Processing Order	The order in which rules are processed, the lower the value, the earlier the rule is processed.
Description	An optional description of the rule.
Client Number Mask	The client number(s) this rule applies to.
Jobname Mask	The job names this rule applies to.
Username Mask	The users this rule applies to.
Message Class	The type of message; it can be found in the log, usually the third column.
Message Number	The number of the message; it can be found in the log, usually the fourth column.
Message Type	The type of the message, S for strings and E for errors, for example.
Message Text	The text of the message, it can be found in the log.
Return Code	The return code to use when the rule matches.
Create Job Note	Should a job note be created when this rule is applied?

Integrating SAP Monitors

Use

With SAP CPS, multiple monitors from different SAP Systems can be watched from one central location. Any monitor or monitor set available in transaction **RZ20** can be imported with the job definition *SAP_ImportCcms-Monitors*. The *XAL* tab of the *SAP System* needs connection credentials of an RFC user with sufficient privileges to use the XAL interface of the given SAP System. The required privileges are outlined in Reference SAP RFC User Privileges.

When you submit the job definition *SAP_ImportCcmsMonitors*, choose a value for the SAP instance parameter and the available Monitor sets and Monitors are available in the respective parameter fields. These fields auto-complete, a character sequence of the name can be entered to reduce the list.

Note

This functionality is only available when the *XAL* tab of the *SAP System* repository object has been filled.

The following job definitions are available for handling monitors:

- *SAP_ImportCcmsMonitors*
- *SAP_DeleteCcmsMonitors*

You may set a maximum setting for the number of RFC connections, please see the tuning for more information.

Note

XBP 3.0 requires a logical system for the client, see Setting Up Logical Systems [http://help.sap.com/-saphelp_nw70ehp1/helpdata/en/b4/b0b13bb3acef3ce10000000a11402f/frameset.htm] for more information on creating logical systems for clients.

Prerequisites

- Client / username and password of the RFC user.
- The SAP System must have a default client configured, please refer to the SAP documentation.

Procedure

To set up the interface to import and delete SAP Monitors:

1. Choose *Environment* > *SAP Systems* in the Navigation bar.
2. Choose *New* from the context menu.
3. On the *SAP System* tab, fill the *Name*, *Connect String* and the *Time zone* fields.
4. On the *XBP* tab, fill in *Client Number*, *Username*, *password*, and check *Default Client*, if no default client has been previously specified.
5. On the *XAL* tab, fill in *Client Number*, *Username*, *password*
6. Choose *Save & Close* to save the changes and close the edit window.
7. Choose *Environment* > *Process Servers* in the Navigation bar.
8. Choose *Start* from the context menu of the process server belonging to the relevant SAP System.

To import a CCMS Monitor

1. Navigate to *Definitions* > *Job Definitions*.
2. Choose *Submit* from the context menu of the job definition named *SAP_ImportCcmsMonitors*.
3. In the submit wizard, fill in the details for Monitor set and Monitor.
4. Choose *Next* and choose the queue of the SAP System.
5. Choose *Submit* to submit the job.
6. Navigate to *Monitoring* > *Monitors* to check the status of the database.

Values

Information about the required configuration data that needs to be specified for integrating SAP monitoring data can be found in the Configuration Details for Integrating SAP Monitors section of the documentation.

Example

You need to be able to monitor the Oracle database of your production NetWeaver AS ABAP+JAVA system.

The following information is known to you:

- The Client is 200, username is masalan and password is masalan.
- The SAP R/3 connect string is: `ASHOST=169.254.10.12 SYSNR=52 TRACE=0`
- XBP version supported is XBP 2.0 (INITXBP2 has run).
- The default client of the SAP System is client 200.
- The monitor set and monitor that you use to monitor the Oracle database in **RZ20**.

First you need to set up or modify the connection to the SAP System; it requires XAL settings.

Tab	**Field**	**Value**
SAP System	Name	sapbw_prd
SAP System	Connect String	`ASHOST=169.254.10.12 SYSNR=52 TRACE=0`
SAP System	Job Handling	All
SAP System	Time zone	Europe/Berlin
SAP System	Application	SAPBW
SAP System	description	Production SAP BW system in Berlin
XAL	Client Number	200
XAL	Username	masalan
XAL	Password	******* (masalan in this example)
XAL	Audit Level	0 - Change actions
XAL	SAP Language	EN
XAL	Description	
XAL	Max RFC Connections	
XAL	Interface Version	automatic

Once the SAP System has been set up, you need to import the SAP Monitors into SAP CPS, you do this with the job definition *SAP_ImportCcmsMonitors*.

1. Navigate to *Definitions* > *Job Definitions*.
2. Choose *Submit* from the context menu of the job definition named *SAP_ImportCcmsMonitors*.
3. In the submit wizard, fill in the details for Monitor set and Monitor.
4. Choose *Next* and choose the queue of the SAP System.
5. Choose *Submit* to submit the job.

6. Navigate to *Monitoring* > *Monitors* to check the status of the database.

Configuration Details for Integrating SAP Monitors

Use

XAL is the interface for retrieving information from the SAP CCMS monitoring infrastructure. The following tabs need to be configured to allow you to integrate SAP Monitors into your SAP CPS Performance Monitor and Dashboard.

First you need to configure connection information to the ABAP stack of the SAP instance.

Setting up Connection Information

See the Connection Details for the ABAP Stack section of the documentation for details on how to set up a connection to the ABAP stack.

Setting up XAL Connection Information

This information is used to retrieve monitoring data from your SAP instance.

Tab	Field	Description
XAL	Client Number	SAP client number as a three digit number.
XAL	Username	The RFC user name. This user needs certain authorizations in the SAP System, as described below in the Reference section under Required Authorizations of the Remote System User. Of course it is possible to use the same user as for XBP and XMW. In this case the user needs all authorizations required by these interfaces.
XAL	Password	Password of the RFC user.
XAL	Check Connection Settings	Allows you to check that the *RFC Connect String*, *Client Number*, *Username* and *Password* are correct.
XAL	Enabled	Allows you to disable the rule, preserving the settings for future use.
XAL	Audit level	The XAL audit level (the audit log is visible via SAP transaction **RZ15**) of the interface, and is an integer with the value range of 0-3, with higher numbers denoting more verbose audit level; only to be used when SAP Support instructs you to do so.
XAL	SAP Language	The that the SAP process server uses to connect to the SAP System.
XAL	Description	The description of the rule. If this is not set the description is <Client Number>/<Username>.
XAL	Max. of RFC connections	Specifies how many RFC connections the SAP process server opens for this client. If omitted, the number of connections is not limited. The SAP process server internally uses a connection pool, so the open connections are reused, and also closed if they are idle for a certain amount of time. This setting can be left empty as it is only of minor importance, since only one RFC connection per SAP process server is opened for the XAL interface.
XAL	Interface Version	The version of the XAL interface that the SAP process server should use. If this field is left on *automatic*, the highest available interface version is used.

Once configured, monitors of different monitor sets in the SAP CCMS monitoring infrastructure can be retrieved and stored in the Monitor tree. This is done via the standard job definitions SAP_ImportCcmsMonitors and SAP_DeleteCcmsMonitors.

Sending Monitoring Data to SAP Systems

Use

Monitoring data from SAP CPS can also be sent to SAP Systems and displayed alongside other SAP-specific monitors in transaction **RZ20** or Solution Manager using the XMW interface. This interface is configured per SAP System and requires an RFC user with sufficient privileges, as outlined in Reference SAP RFC User Privileges.

Before you can send monitoring data to an SAP System about queues and process servers in SAP CPS you will have to create a monitor tree if that has not already been done.

You may set a maximum setting for the number of RFC connections, please see the tuning for more information.

Note

XBP 3.0 requires a logical system for the client, see Setting Up Logical Systems [http://help.sap.com/saphelp_nw70ehp1/helpdata/en/b4/b0b13bb3acef3ce10000000a11402f/frameset.htm] for more information on creating logical systems for clients.

Prerequisites

- An RFC user with sufficient privileges
- A SAP CPS Monitor tree
- An SAP NetWeaver AS 6.20 system with Support Package 12 or higher.

Procedure

To setup synchronization of monitoring data with an SAP System

1. Navigate to *Environment* > *SAP Systems*
2. Choose *Edit* from the context menu of an SAP System.
3. Choose the *XMW* tab and fill in the credentials of the RFC user.
4. Choose *Save & Close*
5. Navigate to *Environment* > *Process Servers*
6. Choose *Stop* and then choose *Start* from the context menu of the corresponding process server. Please wait for the status of the process server to reach *Shutdown* prior to starting it.

To read the SAP CPS monitoring data in an SAP System

1. Log on to the central monitoring system. Open transaction **RZ20**.
2. Open *monitor System / All Monitoring Segments / All Monitoring Contexts* of monitor set *SAP CCMS Technical Expert Monitors*.
3. Look for an MTE of MTE class RedwoodContext.
4. Choose the top node.
5. Choose *Edit* > *Selections* > *Select* > *Node (MTE)* of a class.

6. Enter RedwoodContext.
7. Choose *ENTER* twice. The correct MTE is now selected (but still hidden).
8. Select *Tree* > *Expand tree* > *Along selection*. Now you can see the SAP CPS monitoring data tree, the name of the main node follows this syntax:
    a. <product> <SAP System Name in the Scheduler>@<sap id>
        i. <product> is either SAP CPS or Scheduler, depending on your license.
        ii. <system id> is `SAP:<SID>_<Instance number>` on NetWeaver, `<hostname>` on Redwood Platform
        iii. Example: `Scheduler TB7@sol05` or `SAP CPS TB7@SAP:RIU_09`

Tip

You can also create a new monitor showing SAP CPS data only. This is described in detail in Editing Monitors and Monitor Sets [http://help.sap.com/saphelp_nw70/helpdata/EN/08/-1f5a3b0013ae73e10000000a11402f/frameset.htm]

Values

Configuration details for sending SAP CPS monitoring data to an SAP System can be found in the Configuration Details for Sending Monitoring Data to SAP Systems section of the documentation.

Example

- The Client is 200, username is masalan and password is masalan
- The SAP R/3 connect string is: `ASHOST=169.254.10.12 SYSNR=52 TRACE=0`
- XBP version supported is XBP 2.0 (INITXBP2 has run).
- The default client of the SAP System is client 200.

Tab	Field	Example value
SAP System	Name	sapbw_prd
SAP System	Connect String	`ASHOST=169.254.10.12 SYSNR=52 TRACE=0`
SAP System	Job Handling	All
SAP System	Time zone	Europe/Berlin
SAP System	Application	SAPBW
SAP System	description	Production SAP BW system in Berlin
XMW	Client Number	200
XMW	Username	masalan
XMW	Password	******* (masalan in this example)
XMW	Audit Level	0 - Change actions
XMW	SAP Language	EN
XMW	Description	
XMW	Max RFC Connections	

XMW	Monitor Root	
XMW	Interface Version	automatic

Configuration Details

Use

To be able to send SAP CPS monitoring data to an SAP System, the recipient SAP System must have an ABAP stack with CCMS monitors or Solution Manager, the latter is preferred.

You first need to configure the connection details for the ABAP stack.

Setting up Connection Information

See the Connection Details for the ABAP Stack section of the documentation for details on how to set up a connection to the ABAP stack.

Specifying XMW-specific Settings

Tab	Field	Description
XMW	Client number	SAP client number as a three digit number.
XMW	Username	The RFC user name. This user needs certain authorizations in the SAP System, as described below in the Reference section under Required Authorizations of the Remote System User Of course it is possible to use here the same user as for XBP and XAL. In this case the user needs all authorizations required by these interfaces.
XMW	Password	Password of the RFC user.
XMW	Check Connection Settings	Allows you to check that the *RFC Connect String*, *Client Number*, *Username* and *Password* are correct.
XMW	Audit level	The XMW audit level of the interface, and is an integer with the value range of 0-3.
XMW	SAP Language	The that the SAP process server uses to connect to the SAP System.
XMW	Description	The description of the rule. If this is not set the description is <Client Number>/<Username>.
XMW	Max. RFC connections	Specifies how many RFC connections the SAP process server opens for this client. If omitted, the number of connections is not limited. The SAP process server internally uses a connection pool, so the open connections are reused, and also closed if they are idle for a certain amount of time. This setting can be left empty as it is only of minor importance, since only one RFC connection per SAP process server is opened for the XMW interface.
XMW	Interface Version	The version of the XMW interface that the SAP process server should use. If this field is left on *automatic*, the highest available interface version is used.

Controlling Communication Channels with SAP CPS via PI/XI

Use

You configure SAP CPS to control communication channels by creating a user in the NetWeaver AS JAVA with the required authorizations and use this user in SAP CPS to connect to the SAP instance via the PI interface.

Note

The SAP PI/XI interface requires a Chargeable license.

When you create or edit an SAP System, the required services are added automatically to the process server, so there is no need to change services on a process server.

Prerequisites

- You switched to external control in the Communication Channel Monitor [http://help.sap.com/saphelp_nw04/-helpdata/EN/44/2a1a8620323f0ee10000000a114a6b/content.htm].
- You created a user in NetWeaver, in Visual Administrator for example, and assigned the `xi_af_channel_admin_display` and `xi_af_channel_admin_modify` roles to the `sap.com/com.sap.aii.af.app*AdapterFramework` application for the user.

Procedure

Configure SAP CPS to control the communication channels:

1. Navigate to *Environment* > *SAP System*.
2. Choose *Edit* from the context menu of the SAP System.
3. On the *J2EE* tab, select *PI* and enter the username and password of the user you created previously in SAP NetWeaver.
4. Choose *Save & Close*.
5. Navigate to *Environment* > *Process Servers* and restart the process server belonging to the SAP instance.

See Also

- SAP Communication Channels with PI/XI
- Importing Communication Channels via PI/XI

Monitoring SAP CPS Availability in CEN

Use

You can use the Generic Request and Message Generator (GRMG) to monitor the availability of technical components and entire business processes. GRMG is suitable both for technical monitoring and for application monitoring. The following SAP CPS components can be monitored with GRMG:

- Dispatcher - the process that distributes the load of jobs across the Central Scheduler
- Queues - load-distribution component of SAP CPS
- Process servers - component that executes jobs
- Services of process servers - component of a process server that run jobs, each process server has a service per job definition type

The following syntax is used in transaction **GRMG** to monitor the components mentioned previously, the host and port parameters refer to the host name and connection port of the Central Scheduler.

```
http://<host>:<port>/scheduler/monitoring?[parent=<parentObjectName>&]object=<ObjectType>&name=<ObjectName>
```

Examples:

- `http://<host>:<port>/scheduler/monitoring` Dispatcher
- `http://<host>:<port>/scheduler/monitoring?object=processserver&name=CRM_ProcessServer` Process server *CRM_ProcessServer*
- `http://<host>:<port>/scheduler/monitoring?object=queue&name=CRM_Queue` Queue *CRM_Queue*
- `http://<host>:<port>/scheduler/monitoring?parent=CRM_ProcessServer&object=psservice&name=SAPR3Service` SAPR3 service of the process server *CRM_ProcessServer*

Prerequisites

- A functional Central Monitoring System (CEN) has been configured
- GRMG must have been previously set up and functioning on the Central Monitoring System (CEN)
- Login credentials to the CEN with sufficient privileges to set up a GRMG-lite scenario are required
- The full URL to the Central Scheduler must be known

Procedure

Create a scenario

1. Open transaction **GRMG** on the CEN
2. Choose *Edit/Delete* and then *Create GRMG-Lite Scenario, choose* Yes *to* Create a technical ping to test a URL"
3. Fill in a *Scenario name* and *Description* for the scenario, as well as a *Component name* and *Description* for the Component.
4. Choose *Create*
5. In the *URL/Destination* fill in on of the above URL's, depending on the component you would like to monitor.
6. Fill in *Connect Settings*
7. Select the scenario and choose *Start Scenario* from the *Start* Button.
8. Make sure the scenario is listed as running.

Result

In transaction **RZ20** under the GRMG Monitor set, you should see your SAP CPS-specific monitors you created.

Configuring the Java Scheduler Interface

Use

This section covers the installation of the Java eXternal Batch Processing (JXBP) interface to the SAP NetWeaver Java Scheduler (NWJS), which allows you to schedule Message-Driven Beans.

The SAP NetWeaver Java Scheduler is a simple 'cron-like' scheduler built into the SAP NetWeaver Application Server, as part of version 7.10 (including the Java EE 5 Preview Edition). This scheduler allows specially written Message Driven Beans (MDB's) to be scheduled using simple expressions. It also has an external interface (JXBP) that can be used by more powerful schedulers (like SAP CPS) to achieve the following:

- monitor jobs scheduled by NWJS
- submit new jobs to NWJS for immediate execution.

This allows NWJS to be monitored as part of a larger system, or for some or all NWJS scheduled jobs to be transferred to an external scheduler. The first scenario is most useful for smaller systems, the other scenarios allow the external scheduler to control NWJS jobs. This means that NWJS jobs can be run on a different schedule, more appropriate to the system usage, or scheduled as part of more complex processing, in a job chain for example.

Once the JXBP interface is configured, a process server and queue is created for you automatically with all required services. When you edit an SAP System, the required services are added automatically to the process server, so there is no need to change services on a process server.

Prerequisites

- Appropriate version of SAP NetWeaver Application Server, and the latest NetWeaver Java Scheduler.
- the URL of the JXBP server, aka the server running the NetWeaver Java Scheduler
- A user in the NetWeaver Java Scheduler to be used for JXBP.

Gather the appropriate configuration details

You will need a system running an SAP NetWeaver Application Server Java. You need the following configuration information:

- The hostname of the server (for example java1.masalan.com). This will be referred to as <jxbp hostname>
- The HTTP port number used to connect (for example 50000, 56000, ...). This will be referred to as <jxbp http>
- The username and password for the SAP 'Administrator' user of this system.

Ensure you have the appropriate version of SAP NetWeaver Application Server

While SAP CPS can run on version 7.00 (NetWeaver 2004s) and later, the NetWeaver Java Scheduler and JXBP require version 7.10 and above. The instructions here cover configuring version 7.10. If you are using the Java EE 5 Preview you should contact SAP for an update to JXBP that allows external schedulers to connect to it.

Procedure

Configure a user in the NetWeaver Java Scheduler to be used for JXBP.

1. Navigate to NetWeaver Administrator (http://<jxbp hostname>:<jxbp http>/nwa)
2. Navigate to *Operation Management* > *Jobs* > *Java Scheduler* > *External Schedulers*.
3. Choose *Add* and fill in the details of the <jxbp user> and choose *Add*. These details will be needed when you connect SAP CPS to this Java Scheduler.

Note

If you do not see the appropriate page here, you may not have a version of the NetWeaver Java Scheduler with JXBP. Contact SAP support for help.

Note

Redwood recommends the use of *redwood* for name and user name.

Note

TimeoutGracePeriod should be set to one week, or 604800000.

To set up the interface to interact with the Java Scheduler:

1. Choose *Environment* > *SAP Systems* in the Navigation bar.
2. Choose *New* from the context menu.
3. On the *SAP System* tab, fill the *Name*, *J2EE URL* and the *Time zone* fields.
4. On the *J2EE* tab, fill in the *Username* and *password* in the fields, choose *Check connection* to make sure the account details are valid.
5. Choose *Save & Close* to save the changes and close the edit window.
6. Choose *Environment* > *Process Servers* in the Navigation bar.
7. Choose *Start* from the context menu of the process server belonging to the relevant SAP System.

Starting the SAP CPS RFC Server

Use

The SAP CPS RFC server allows ABAP stacks to connect to SAP CPS and interact with the scheduler. The SAP CPS interfaces with the SAP Applications API. At the time of this writing, the following SAP components have been optimized for SAP CPS RFC server (SAP Applications API):

- Solution Manager
- Closing Cockpit

Note

More SAP components are currently being optimized to make use of the RFC server.

Note

You need an ABAP stack of NetWeaver 7.00 SPS 16 or later to connect to the RFC server.

You start the RFC server by setting the process server parameter *SAP_EnableRfcServer* to `true` on the process server that belongs to the SAP System which will connect to the RFC server. You can connect to multiple SAP CPS process servers to a single SAP instance and you can enable an RFC server for each of them, however, when this is the case you should set the process server parameter *SAP_API_DefaultScheduler* for each, it should be set to `false` on all SAP process servers except one, which is the default scheduler for the SAP Applications API of that SAP instance. The value of this scheduler parameter is set to `true` by default.

Security

The RFC server has to be enabled manually, it is not enabled by default. ABAP users are mapped to SAP CPS users and need the `SAP_BC_BATCH_ADMIN_REDWOOD` role in the SAP R3 subsystem and `Business`

`Automation User` for the SAP System in SAP CPS. The SAP CPS user needs to be have the same isolation group as the SAP System, or login will be denied.

Prerequisites

Variable	**Description**
<username>	The username of the RFC user that SAP CPS uses to connect to the SAP System
<isolationgroup_partition>	The partition of the isolation group, usually `GLOBAL`.
<isolationgroup>	The name of the isolation group, usually `GLOBAL` when it has not been set.
<SAP CPS_server>	The FQDN of the SAP CPS server.
<SAP_client>	The SAP client used by the SAP CPS RFC user.

Procedure

Enabling the RFC server in SAP CPS

1. Navigate to *Environment* > *Process Servers*
2. Choose *Edit* from the context menu of the process server belonging to the SAP Instance you want to connect from.
3. On the *Parameters* tab, choose *New*.
4. Fill `SAP_EnableRfcServer` into the *Name* field and `true` into the *Value* field.
5. If there are more process servers with the RFC server enabled connecting to this instance:
   a. Choose *New*, fill `SAP_API_DefaultScheduler` into the *Name* field, and `true` or `false` into the *Value* field, only one SAP API default scheduler is allowed per SAP instance.
6. Choose *Save & Close*.
7. Choose *Stop* from the context menu of the process server, wait for the process server to be marked as *Shutdown* in the ui.
8. Choose *Start* from the context menu of the process server.

Creating an RFC Destination to SAP CPS

1. Open transaction **SM59**.
2. Choose *Create* from the *Edit* menu.
3. Fill in a name into the *RFC Destination* field.
4. Fill `T` into the *Connection Type* field.
5. Fill in a description.
6. On the *Technical Settings* tab choose *Registered Server Program*.
7. Fill `<username>-<isolationgroup_partition>.<isolationgroup>@<SAP CPS_server>-<SAP_client>`

8. Fill in the *Gateway Host* and *Gateway service* fields with name of the gateway and the service of the current SAP instance.

9. Save, choose *Test Connection* and wait for the result, if the result is OK, you have successfully configured the connection.

Integrating Into Closing Cockpit

Use

Closing cockpit lets you organize you closing procedures and have the process followed in real time, all automated tasks are carried out with a minimal delay, when a task requires user intervention, the user can be alerted from within closing cockpit via email, for example, and the alert can be escalated if it is not resolved in a timely manner. This allows fluid closing processes and greatly reduces the time it takes to close the books. SAP CPS is the scheduler which organizes and triggers all automated tasks in time across your system landscape, which can include SAP and non-SAP Systems. For example, when you previously had to run a job on a Unix system to populate a database with data after which the SAP team could proceed with the launch an ABAP to process that data the time between the two steps could be quite long; with SAP CPS the time between those two tasks can be automated and reduced to a second or less. When you have setup Closing Cockpit, you can follow SAP CPS Closing Cockpit-related tasks in both Closing Cockpit and SAP CPS.

Prerequisites

- SAP Basis release 7.00 with one of the below Support Packages and relative SAP notes (if any):
  - SP 14 plus SAP notes 1112590, 1129030, 1154821
  - SP 15 plus SAP Note 1129030
  - SP 16, no SAP notes needed
- SAP Enhancement Package 3 for SAP ECC 6.0
- A SAP Application connection to the SAP CPS RFC server

Note

Closing Cockpit integration requires the *ProcessServerService.SAP.cc* license key. If you want to be able to send email, you also need the *Module.Mail* license key.

Procedure

Adding a SAP CPS task to a Closing Cockpit task list

1. Open transaction **CLOCOC**.

2. Open your task list or choose *Other Template/Task List* from the *Template/task List* menu and choose a task list.

3. Choose the *Edit* button and make sure the name of your task list is followed by *Change* in the title bar of SAPGUI.

4. Choose *New Task* from the context menu of a folder in the task list hierarchy.

5. In the *Closing Cockpit Display/Create/Change Task* window choose *Remote Task* and select the name of a SAP CPS job definition in the *Remote Task* field (you may select any job definition the SAP CPS user is allowed to use).

6. Fill in a *Queue* and parameters, if there are any, just as you would in SAP CPS.

Note

If you do not have the *Remote Task* field, then you do not have correctly set up a connection to a SAP CPS RFC server.

Note

You can also create a task list template, import one from Closing Cockpit or Solution Manager, or create a task list. That is out of the scope of this guide. Please see the SAP related documentation for more information on closing cockpit.

Integrating into Solution Manager

Use

SAP Solution Manager is a free component with SAP NetWeaver. You can schedule jobs in SAP CPS from within Solution Manager via the SAP Applications API. Before you can schedule jobs, you need to set up the RFC server in SAP CPS and create a RFC destination pointing to SAP CPS, this is outlined in the Starting the SAP CPS RFC Server section of the documentation. The Solution Manager integration allows seamless change management, which is built into SAP Solution Manager and enriches the it with the functionality of SAP CPS; running processes anywhere across your data center across platforms. SAP CPS sends the status information of the job back to SAP Solution Manager and integrates into its scenarios.

For more information on SAP Solution Manager integration, see Editing Monitors and Monitor Sets [http://-help.sap.com/saphelp_sm40/helpdata/en/c3/44390086414b4f81a10773b7a11f80/frameset.htm].

Note

See SAP Note 1122497 and SAP Note 1265635 for more information.

Managing Roles and Users

Use

SAP CPS features a simplified security model based on built-in Redwood roles. Users and roles from external authentication systems are used to access the user interface, it is not possible to create users or roles inside SAP CPS. Basic privileges can be granted inside the authentication system, via the built-in roles; the process depends on the authentication system you are using. Fine-grained privileges can then be granted in the user interface to custom roles that you previously created in the external authentication system.

This section contains the following topics:

1. Isolating Users and Objects
   a. Partitions
   b. Creating Isolation Groups and Assigning them to Users
2. Security in SAP NetWeaver
   a. Creating Users in SAP NetWeaver
   b. Creating Roles in SAP NetWeaver

3. Assign Users from External Platforms to Jobs

4. Privileges

   a. System Privileges

      i. Granted System Privileges

   b. Object Privileges

   c. Granting and Revoking System Privileges

   d. Granting and Revoking Object Privileges

5. Privileges Required to use objects

Isolating Users and Objects

Use

With an isolation group, a user is limited to interacting with objects of the same isolation group. Isolation groups are linked to partitions and to the *ProcessServerService.SAP.limit* license key. Every isolation group requires its shared partition. When this license key is set to `0`, you can create as many isolation groups as you want.

Every user has an isolation group. A user with an isolation group can only see users and objects of his enabled isolation group or the `GLOBAL` isolation group, provided he has the required system and/or object privileges. Objects inherit the isolation group of the partition in which they are stored. Regardless of privileges, a user will never be able to see or access an object with a `non-GLOBAL` isolation group if he does not have the same isolation group, enabled for the session, as the object.

This means that, when you log on, you choose your enabled isolation group for the session (if you have more than one) and this isolation group allows you to access objects which have the `GLOBAL` isolation group or the same one as your enabled one.

To create isolation groups, refer to the Assign Isolation Groups section of the documentation.

Partitions

Use

Partitions are used to store objects in the repository and isolate them from one-another together with isolation groups. Partitions are known as shared when they have an isolation group that is null.

If the *ProcessServerService.SAP.limit* license key is set to `0`, the following restrictions are in place:

- one SAP System may be created per partition with one XBP client.
- any partition with a `null` isolation group is read-only.
- only the system can create partitions with a `null` isolation group.
- all monitors from SAP (vi XAL) will be created in the appropriate isolation group.
- SAP batch event will raise events in the isolation group only.
- job control rules will function only for jobs from the defined XBP client, child jobs will be picked up regardless in which client they were created.

- variant management will be limited to the defined XBP client.
- BW restart will be possible, however, it requires the ProcessServerService.SAP.ProcessChainRestart license key
- Mass activities will be possible, however, they require the ProcessServerService.SAP.MassActivities license key
- Redwood transports will be available for mass activities
- SAP Applications API will be available, however, they require the ProcessServerService.SAP.ApplicationsAPI license key
- SAP_AbapRun, SAP_BW_ProcessChainRun and all the other SAP job definitions that run jobs will only allow ABAP reports, variants, and process chains that have been imported into SAP CPS

Managing Users and Roles in External Systems

Use

Users and roles are created in external systems, how you proceed depends on the authentication system you use:

- Security in SAP NetWeaver

See Also

- Managing Users and Roles in SAP CPS

Security in SAP NetWeaver

Use

The scheduler integrates with UME to determine which users have at least one of the following two actions:

- redwood.com/Scheduler.AccessScheduler - access only.
- redwood.com/Scheduler.ManageScheduler - manage the scheduler.

Roles with these actions are managed in UME. The following two sections illustrate how to assign roles and isolation groups to users:

- Assigning Isolation Groups
- Assigning Roles

As soon as users with one of the above actions log on for the first time, objects are automatically created. Any role that has any of the above actions, will be created in SAP CPS as soon as a user with that role logs on. The isolation administrator can also add users from UME. Authorizations are synchronized at login-time, so changes will be effective upon the next login.

Coarse-grained permission assignment is managed in UME, by creating a role matching the following role names and assigning it the *redwood.com/Scheduler.AccessScheduler* action.

- `scheduler-administrator` - can perform all actions (requires `scheduler-user`).
- `scheduler-event-operator` - can raise and clear events (requires `scheduler-user`).

- `scheduler-job-administrator` - can create/edit/delete event definitions, job definitions, job chains. Can modify jobs (requires `scheduler-user`).
- `scheduler-user` - has access to SAP CPS only, cannot see any objects.
- `scheduler-viewer` - read only access to all objects (requires `scheduler-user`).
- `scheduler-isolation-administrator` - can create/edit/delete isolation groups and add users to these (requires `scheduler-user`).
- `scheduler-screen-reader` - indicates that you are using a screen reader (requires `scheduler-user`, not displayed when running in SAP NetWeaver).

Fine-grained permission assignment can be done inside SAP CPS by assigning permissions to users or custom roles created in UME. It is not possible to change the above roles from within SAP CPS.

See Also

- Isolation Groups
- System Privileges
- Object Privileges

Creating Users in SAP NetWeaver

Use

User objects represent users of the system and can have roles granted to them in order to have privileges to perform actions.

Users are part of the Security section in the navigation bar. The following standard users are provided with SAP CPS:

- *Redwood System* - a user that represents the system.

Users are imported automatically from the SAP User Management Engine (UME) as soon as they log on for the first time and kept in sync with UME at each login. This synchronization only occurs one way, from UME to SAP CPS. Users are created in UME and cannot be created in SAP CPS. They must have the following action assigned to at least one of their roles to get read-only access:

- *redwood.com/Scheduler.AccessScheduler*

Administrator users additionally have the UME role `scheduler-administrator` or the following action assigned to one of their roles:

- *redwood.com/Scheduler.ManageScheduler*

To view all the available users, choose *Security* > *Users* in the Navigation bar.

You can enable and disable login for a user; when you disable login for a user, the user will be notified at the next login attempt that the user has no access to the system.

Prerequisites

- A user account and password with appropriate privileges for managing users in the SAP NetWeaver system where SAP CPS was installed.

- The URL to the *User Management interface* of the SAP NetWeaver system where SAP CPS was installed.

Procedure

Create a user in SAP NetWeaver UME

1. Connect to the *User Management interface* in SAP NetWeaver where SAP CPS was installed.
2. Choose *Create User* and fill in the data.
3. Add this user to at least one of the roles created during the installation of SAP CPS.
4. Log on with this user in SAP CPS.

Disable a user from logging into SAP CPS

1. Navigate to *Security* > *Users*.
2. Choose *Disable Login* from the context menu of the user

Values

- *Name* - The name of the user.
- *Description* - The description of the user.
- *Comment* - A comment about the user.
- *Creation Time* - The time the user ID was created.
- *Last Modification time* - The last time the user ID was modified.

Creating Users in SAP NetWeaver

Use

User objects represent users of the system and can have roles granted to them in order to have privileges to perform actions.

Users are part of the Security section in the navigation bar. The following standard users are provided with SAP CPS:

- *Redwood System* - a user that represents the system.

Users are imported automatically from the SAP User Management Engine (UME) as soon as they log on for the first time and kept in sync with UME at each login. This synchronization only occurs one way, from UME to SAP CPS. Users are created in UME and cannot be created in SAP CPS. They must have the following action assigned to at least one of their roles to get read-only access:

- *redwood.com/Scheduler.AccessScheduler*

Administrator users additionally have the UME role `scheduler-administrator` or the following action assigned to one of their roles:

- *redwood.com/Scheduler.ManageScheduler*

To view all the available users, choose *Security* > *Users* in the Navigation bar.

You can enable and disable login for a user; when you disable login for a user, the user will be notified at the next login attempt that the user has no access to the system.

Prerequisites

- A user account and password with appropriate privileges for managing users in the SAP NetWeaver system where SAP CPS was installed.
- The URL to the *User Management interface* of the SAP NetWeaver system where SAP CPS was installed.

Procedure

Create a user in SAP NetWeaver UME

1. Connect to the *User Management interface* in SAP NetWeaver where SAP CPS was installed.
2. Choose *Create User* and fill in the data.
3. Add this user to at least one of the roles created during the installation of SAP CPS.
4. Log on with this user in SAP CPS.

Disable a user from logging into SAP CPS

1. Navigate to *Security* > *Users*.
2. Choose *Disable Login* from the context menu of the user

Values

- *Name* - The name of the user.
- *Description* - The description of the user.
- *Comment* - A comment about the user.
- *Creation Time* - The time the user ID was created.
- *Last Modification time* - The last time the user ID was modified.

Creating Roles in SAP NetWeaver

Use

A role is a set or group of privileges that can be granted to users or other roles. Roles are part of the *Security* section in the navigation bar.

The following standard roles are provided with SAP CPS:

- `scheduler-administrator` - can perform all actions, except manage isolation groups (requires `scheduler-user`)
- `scheduler-event-operator` - can raise events (requires `scheduler-user`)
- `scheduler-job-administrator` - can create/edit/delete event definitions, job definitions, job chains and modify jobs (requires `scheduler-user`)
- `scheduler-user` - has access to SAP CPS only, cannot see any objects

- `scheduler-viewer` - read only access to all objects (requires `scheduler-user`)
- `scheduler-isolation-administrator` - can create/edit/delete isolation groups and add users to these (requires `scheduler-user`)
- `scheduler-screen-reader` - indicates that you are using a screen reader (requires `scheduler-user`, not displayed when running in SAP NetWeaver)

These are the standard roles, it is not possible to create roles directly in SAP CPS, or edit the standard roles. You create a role with the corresponding name in the UME, assign it the *redwood.com/Scheduler.AccessScheduler* action and grant it to SAP CPS users. Once you grant these roles to users in the UME, the users will get the corresponding SAP CPS roles granted and have the privileges granted to these roles as soon as they log on or the isolation administrator adds them to an isolation group.

The UME also allows you to grant SAP CPS privileges directly to roles via the following actions:

- *redwood.com/Scheduler.AccessScheduler* - read-only access, all roles that you want to use in SAP CPS must have this action assigned.
- *redwood.com/Scheduler.ManageScheduler* - scheduler Administrator (same as `scheduler-administrator` role)
- *redwood.com/Scheduler.ManageSchedulerIsolation* - isolation group administrator (same as `scheduler-isolation-administrator` role); you are not allowed to grant the *redwood.com/Scheduler.ManageScheduler* action to the same role.

SAP strongly recommends you start by creating the above roles in the UME and assign them the *redwood.com/Scheduler.AccessScheduler* action. Once that is done, you grant these roles to users. If the roles do not exactly suit your needs, create custom roles (with names that do not match the above roles) in the UME and assign them the same action; these roles will be editable in SAP CPS and allow you to extend privileges of users.

Accessibility

SAP CPS has been tested with the *JAWS* screen-reader for the visually impaired. When you want to use a screen-reader, you have the following options, depending on your application server:

- SAP NetWeaver - SAP UME allows you to set a property for visually impaired users, this property is supported by SAP CPS
- Other supported application servers - you may use the `scheduler-screen-reader` role or append `&sap-accessibility=X` in the URL. Note that the use of the role is highly recommended. Despite what the name might suggest, the URL parameter is supported on all supported application servers, and is designed for testing purposes.

Granting Privileges

You can grant privileges to custom roles. Custom roles are recognizable by the name (it is not allowed to be one of the above) as well as the description and comment fields, which contain *Created automatically on first login*.

Some privileges are not dependent on a partition or isolation group, these are known as global Privileges.

You grant specific privileges to roles for each object type; available privileges are dependent on the object. These privileges are known as system privileges and are granted via the role object. There are two levels at which you can grant system privileges, *Access* and *Admin*. Privileges granted with the *Admin* level allow the grantee to grant the privileges to other users.

You can also grant privileges for specific objects on object-level to users or roles, these are known as object privileges and are granted via the *Security* tab on the object.

The following topics in the Security Guide and Administration Guide provide more information on privileges and how you grant them:

- System Privileges
- Object Privileges
- Granting and Revoking System Privileges
- Granting or Revoking Object Privileges

Prerequisites

- The URL to the NetWeaver UME. Usually http://<host>:<port>/useradmin
- An account with sufficient privileges to create a role and add members to the role.
- The username and password of a user you add to the role.
- A Scheduler Manager user account for SAP CPS

Procedure

Creating a role to map to a SAP CPS role

1. Connect to the *UME* in SAP NetWeaver where SAP CPS was installed.
2. Choose *Role* in the drop-down box for *Search Criteria*.
3. Choose *Create Role* and enter a name of one of the SAP CPS roles into the *Unique Name* field.
4. Choose the *Assigned Actions* tab.
5. Search for `scheduler` and choose the action name *redwood.com/Scheduler.AccessScheduler*, choose *Add* and *Save*.
6. Log out of the *UME*.

Creating a custom role to manage isolation groups in the UME

1. Connect to the *UME* in SAP NetWeaver where SAP CPS was installed.
2. Choose *Role* in the drop-down box for *Search Criteria*.
3. Choose *Create Role* and enter a role name in the *Unique Name* field, SAP recommends to prefix the role name with `scheduler` as it will be easier to distinguish the role fro other roles in the UME .
4. Choose the *Assigned Actions* tab.
5. Search for `scheduler` and choose the action names *AccessScheduler* and *ManageSchedulerIsolation*.
6. Choose *Add* and *Save*.
7. Log out of the *UME*.

Editing a custom role in SAP CPS

1. Navigate to *Security* > *Roles*.
2. Choose *Edit* from the context menu of an editable role.Editable roles have a description: *Created automatically on first login*.
3. On the *Assign Privileges* tab, choose an Object definition and then *Next*.
4. Choose the desired range of the privileges.
5. Choose a *Rank* with the desired privileges. *Admin* privileges allow the user to perform the action and to grant the privilege to others as well. *Access* privileges allow the user to perform the actions.

Assigning a user to a role

1. Connect to the *UME* in SAP NetWeaver where SAP CPS was installed.
2. Fill in the *Search Criteria* field and locate the user.
3. Choose the user, and choose the *Assigned Roles* tab.
4. Search for one of the roles previously created, and choose *Add*.
5. Choose *Add* and *Save*.
6. Log out of the *UME*.

Values

Assign Privileges

- *Object Definition* - The role name.
- *Grantable* - If this option is selected, the user can grant this role to any other user.

Assigned Global Privileges

- *Granted Subject* - The privilege name.
- *Grantable* - If this option is selected, the user can grant this privilege to any other user.

Example

- The URL to the NetWeaver UME http://sapnwprd.masalan.com:50200/useradmin
- The account with sufficient privileges to create a role and add members to the role is Administrator
- The username and password of the user to add to the role is masalan and process1 respectively, created previously.
- The Scheduler Manager user account is ProcAdmin.

Creating a role that maps to the *scheduler-event-operator* SAP CPS role

1. Connect to http://sapnwprd.masalan.com:50200/useradmin using the Administrator login
2. Choose *Role* in the drop-down box for *Search Criteria*.
3. Choose *Create Role* and enter *scheduler-event-operator* into the *Unique Name* field.

4. Choose the *Assigned Actions* tab.

5. Search for scheduler and choose the action name *AccessScheduler*, choose *Add* and *Save*.

6. Log out of the *UME*.

Managing Users and Roles in SAP CPS

Use

Users and roles are defined in external authentication systems, however, you can manage certain SAP CPS-specific aspects of these in SAP CPS. The user or the role must to appear in SAP CPS.

Users

You can disable and enable login for users, this only affects SAP CPS and has no incidence on other applications running on the JEE application server. In isolated environments, users can be members of one or more isolation groups, this can be managed by the isolation group manager.

Roles

Only custom roles, or roles that were not built-in, can be edited.

Actions

Users support the following actions:

Action	Description
Enable login for user	Enable login for a disabled user
Disable login for user	Disable login for an enabled user
Edit	Edit the user's privileges
Expand All	Expand all users in the current filter
Import Users	Import users into an isolation group, you need to be the isolation group manager to be able to import users

Roles support the following actions:

Action	Description
Edit	Edit the role's privileges
Expand All	Expand all roles in the current filter

See Also

- Managing Users and Roles in External Systems

Creating Isolation Groups and Assigning them to Users

Use

With an isolation group, a user is limited to interacting with objects of the same isolation group as well as objects from the `GLOBAL` isolation group. You assign users to isolation groups in the Security section in the navigation bar. The user must either have the role *scheduler-isolation-administrator* or a custom role granted which has the *redwood.com/ManageSchedulerIsolation* action to be able to add users to an isolation group. Users can have multiple isolation groups; in this case, the user needs to choose an isolation group for the session during login.

Note

In the Free of Charge version, only one SAP client can be in an isolation group. This is governed by the value of the *ProcessServerService.SAP.limit* license key.

Prerequisites

- the user needs the role *scheduler-isolation-administrator* or a custom role granted which has the *redwood.com/Scheduler.AccessScheduler* action.

Procedure

Create an isolation group

1. Navigate to *Security* > *Isolation Groups*.
2. Choose *New* from the context menu.
3. Enter a name for the isolation group. Names should be all upper case.
4. Choose *Save & Close* to exit edit dialog.

Assign an isolation group to users

1. Navigate to *Security* > *Isolation Groups*.
2. Choose *Import Users*
3. Use the search dialog to locate users from the Authentication system, wildcard `*` and `?` are allowed.
4. Select the users to import and choose Import.
5. Close the window when you are finished importing users into the isolation group.

Assign Users from External Platforms to Jobs with Credentials

Use

To run jobs on external platforms as a specific user, you can specify the user and password in the job definition *Run As User* field and you can use credentials. Credentials are used to store login details for users on external systems in both a central and secure manner. When the password of a user changes, you can change it on the credential, which takes effect immediately; if you had specified the user explicitly on job definitions, you would have to change the password on every job definition, this is error-prone. Credentials are part of the *Security* section of the navigation bar.

Note

SAP strongly recommends the use of credentials over specifying a username and password in the *Run As User* field.

The following credential protocols are available by default:

- login - the default protocol for external systems
- oracle - the protocol for connecting to Oracle databases, with SQLPLUS job definitions

login Credential

The built-in *login* credential protocol is used to authenticate users against the remote system defined in *Endpoint*.

Note

The value in *Endpoint* must match the value of the *RemoteHostName* process server parameter for the corresponding process server.

oracle Credential

The built-in *oracle* credential protocol is used to authenticate users against the oracle database defined in *Endpoint*. You must add the SQLPLUS job definition type to a platform agent. The `sqlplus` binary must be on the path of the user running the job. On the SQLPLUS job definition you specify `<user>@<endpoint>` to sue the credential *<user>-<endpoint>-oracle*.

Actions

Credentials support the following actions:

Action	Description
Edit	Edit the credential
Edit Security	Edit the security of the credential
Delete	Delete the credential
Expand All	Expand all credentials in the current filter
New	Create a new credential

Standard Credentials

With standard credentials, you specify the username in the *Run As User* field of a job definition. When the job gets allocated to a process server, SAP CPS retrieves the credential by searching for a credential where the following conditions are met:

- the process server parameter *RemoteHostName* matches the value of *Endpoint*
- the value of the *Run As User* matches the value of *Real User*

The *Password* is used in combination with the value of *Run As User* to log on and execute the code in the job.

When either fails, the job is put into status *Error*, as no password could be found for the user on that system.

You use standard credentials, when the usernames and passwords of specific users are the same across systems.

Virtual Credentials

Virtual credentials allow you to specify a user across several systems by the same name although their username and password are different. When the job gets allocated to a process server, SAP CPS retrieves the credential by searching for a credential which meet the following conditions:

- the process server parameter *RemoteHostName* matches the value of *Endpoint*
- the value of the *Run As User* matches the value of the *Virtual User* (the `{Virtual}:` prefix is trimmed)

The *Real User* and *Password* from the credential are the used to log on and execute the code in the job.

When either fails, the job is put into status error, as no username and password could be found for the user on that system.

You use virtual credentials, when the usernames and passwords of specific users are different across systems.

For example, all your systems could have a virtual user named `erpuser` which has a different username and password on one or more systems. In the *Run As User* field of the job definition you can then specify `{Virtual}:erpuser`. The credential with the *Virtual User* `erpuser` will be used, the actual username and password can be different across systems, see the example below.

Syntax

When you need to specify a standard credential in the *Run As User* field of a job definition, provide just the username of the user for Unix and VMS systems. On windows, you might need to specify a windows domain, use *[domain\]username*, for example *masalan\jdoe*.

On Windows, the syntax without credentials for the *Run As User* is *[domain\]username/password*, for example *masalan\jdoe/mysecret* or *jdoe/mysecret*; on UNIX and OpenVMS it is *username*.

Summary

	Standard Credentials	**Virtual Credentials**
Run As User value	username	{Virtual}:vuser
Lookup by	Username=username EndPoint=RemoteHostName	VirtualUser=vuser EndPoint=RemoteHostName
Result	Password	Username Password

Security

To use a credential, the user must have sufficient privileges, like *Credential.view* system privileges or corresponding object privileges.

To create a credential, you need to have privileges to see the credential protocol you want to use.

Privilege	**Description**
Credentials.Create	Create credentials
Credentials.Delete	Delete credentials
Credentials.Edit	Edit credentials
Credentials.View	Access credentials

You can grant privileges on two levels, *Access* and *Admin;* a privilege granted on *Admin* level allows the grantee to grant the privilege to other users. These privileges can be granted *system-wide*, per *partition* or *isolation group*.

You specify which users can access, edit, and delete the credential on the *Security* tab of the credential.

- Granting and Revoking System Privileges
- Granting or Revoking Object Privileges

Note

Please be very careful when you create credentials, make sure that only specific users can use them. A credential allows you to log onto a system without a password and can potentially cause havoc if you are not careful.

Procedure

Create a credential

1. Navigate to *Security* > *Credentials*
2. Choose *New* from the context menu, fill a hostname into the *Endpoint* field (hostname must match the *RemoteHostName* process server parameter).
3. Specify the username and password, which in most cases are case-sensitive.
4. Choose *Save and Close*.

Use a credential

1. Navigate to *Definitions* > *Job Definitions*.
2. Choose *Edit* from the context menu of a job definition that runs on an external platform.
3. On the *Source* tab, enter the username of the user provided in a credential.
4. Submit the job and choose a queue that is served by a process server running on the *Endpoint* specified in the credential.

Values

Field	**Description**
Protocol	The protocol used for the credential, *login* is the default
Endpoint	The hostname of the system where the user is valid; this must be set to the same value as the *RemoteHostName* process server parameter
Real User	The username of the user on the endpoint
Password	The password of the user on the endpoint
Virtual User	The virtual username for the user

Example

The production ERP system at Masalan Inc has an additional development and a quality assurance system; job definitions are developed and tested before they reach production. To ease the pain of migration, Masalan Inc uses the virtual user property of their credentials. The following accounts are used to run the jobs on the different systems:

- `pr1adm` - production system
- `qa1adm` - quality assurance system
- `dv1adm` - development system

Note

The use of administrative SAP users is for illustration purposes only, you should not use these accounts without your SAP Administrator knowing.

In the development system, you create a credential as follows:

Field	Value
Protocol	login
Endpoint	`dv1.masalan.com`
Real User	`dv1adm`
Password	`someGoodSecret`
Virtual User	erpuser

In the quality assurance system, you create a credential as follows:

Field	Value
Protocol	login
Endpoint	`qa1.masalan.com`
Real User	`qa1adm`
Password	`someGoodSecret`
Virtual User	erpuser

In the production system, you create a credential as follows:

Field	Value
Protocol	login
Endpoint	`pr1.masalan.com`
Real User	`pr1adm`
Password	`someGoodSecret`
Virtual User	erpuser

Job definitions that need to run as the environment-specific user contain `{Virtual}:erpuser` in the *Run As User* field; the job definition can be migrated without the need to change the *Run As User* field as in each system, the correct user is used to execute the code.

Credential Protocols

Use

Credential protocols are used to store connection information for remote systems, such as Windows process servers.The following built-in credentials are available:

- login - standard protocol for platform agents
- oracle - standard protocol for accessing Oracle databases

The following protocol can be created if you require mail authentication:

- smtp - protocol used for mail authentication

If you require a specific protocol for your application or platform, protocols can be created/modified for you on a consultancy-basis.

Actions

Credential protocols support the following actions:

Action	Description
Edit	Edit the credential protocol
Edit Security	Edit the security of the credential protocol
Delete	Delete the credential protocol
Expand All	Expand all credential protocols in the current filter
New	Create a new credential protocol
Filter > New Filter	Create a new credential protocol filter
Filter > Edit Filter	Edit current credential protocol filter
Filter > Delete	Delete current credential protocol filter

Finding Credential Protocols

You can search for credential protocols using filters and the *Search Credential Protocols* box on the Credential Protocols tab. This box is known as the *intelliSearch* box and located under your username on the top right-hand side of the user interface. Filters allow you to specify a list of objects with static criteria. IntelliSearch allows you to specify complex queries in a simple way using prefixes. Prefixes are used to specify which property you are searching in and have short and long syntaxes. For example, if you want to display all credential protocols with the term *dev* in the comment, you would use the search criteria as follows:

```
c:dev
```

You can search more than one property, as follows:

```
c:dev n:ORA
```

Note

No spaces should be entered before or after the colon (*:*).

See the Advanced Object Search for more information.

The following table illustrates the available prefixes for credential protocols:

Prefixes	Description
n, name	searches the name property
c, com, comment	searches the comment property
d, desc, description	searches the description property
a, app, application	searches the application property

Deleting Credential Protocols

You can only delete credential protocols when no other objects relate to them. For example, if there are credentials that use the credential protocol, the credential protocol cannot be deleted until all credentials that use it have been modified. You can see all credentials that relate to the credential protocol in *Related Objects* in the lower detail pane and on the show page.

The table in related objects contains three columns:

- Type - the type of object with a link to it
- Related Object - the name of the object with a link to it
- Used As - objects can sometimes be used in different roles

Security

Privilege	Description
CredentialProtocol.Create	Create credential protocols
CredentialProtocol.Delete	Delete credential protocols
CredentialProtocol.Edit	Edit credential protocols
CredentialProtocol.View	Access credential protocols

You can grant privileges on two levels, *Access* and *Admin;* a privilege granted on *Admin* level allows the grantee to grant the privilege to other users. These privileges can be granted *system-wide*, per *partition* or *isolation group*.

You specify which users can access, edit, and delete the credentials on the *Security* tab of the credentials or credential protocols.

- Granting and Revoking System Privileges
- Granting or Revoking Object Privileges

Privileges

Use

There are two types of privileges, object privileges and system privileges. Object privileges can be granted for a specific object. System privileges can be granted for object types in the entire system, in a partition or an isolation group and allow you to limit the privilege to objects in a particular partition or with a particular isolation group.

Isolation groups isolate users and objects from other users and objects. A user must have the same isolation group to be able to see and access users or objects in an isolation group, provided the user has the required privileges. A user can have one or more isolation groups, in this case he must choose his enabled isolation group for the session at logon. See the isolation group section for more information.

Object Privileges

Object privileges always relate to a specific object and allow the grantee a specific right on the object. A *View* privilege on the *RS_PrintStatements* job definition, for example, is only valid for that job definition. If the user has no other job definition-related system privileges and no other object privileges on job definitions, the only job definition the grantee can view, or access, is *RS_PrintStatements*.

Object privileges cannot be granted directly, you grant ranks of privileges. For example, the *Edit* rank contains both *View* and *Edit* privileges, this prevents human error, as you need to see an object before you can edit it. Furthermore, privileges can be granted as *Access* and *Admin*, when you grant a privilege as *Admin*, the grantee can grant the privilege to other users.

System Privileges

System privileges are granted on three levels, per partition, per isolation group or system wide. If you are using multiple partitions or isolation groups, you can restrict a system privilege to one partition or isolation group.

The *EventDefinition.Raise* system privilege, for example, allows the grantee to raise all events he can view, combined with the *EventDefinition.View* he can access all events in a partition, isolation group, or across the entire system.

The default roles cannot be edited, but roles you created in external authentication systems are editable in SAP CPS provided you have the necessary security module, please check your license if you are unsure. The default permissions granted to built-in roles are listed in the Granted System Privileges section.

See Also

- Granting and Revoking System Privileges
- Granting and Revoking Object Privileges
- Privileges Required to use Objects

System Privileges

Use

Granted System Privileges

Use

The following pre-defined roles are available:

- `scheduler-administrator` - can perform all actions.
- `scheduler-event-operator` - can raise and clear events.
- `scheduler-job-administrator` - can create/edit/delete event definitions, job definitions, job chains. Can modify jobs.
- `scheduler-user` - has access to SAP CPS only, cannot see any objects.
- `scheduler-viewer` - read only access to all objects.
- `scheduler-isolation-administrator` - can create/edit/delete isolation groups and add users to these.
- `scheduler-screen-reader` - indicates that you are using a screen reader (not displayed when running in SAP NetWeaver).

All users need at least the *scheduler-user* role, even if they have the *scheduler-administrator* role, or they will not be able to access SAP CPS.

System privileges can be granted for the entire system, a partition or an isolation group, this allows you to limit the privilege to objects in a particular partition or with a particular isolation group.

System-wide privileges are only valid in the partitions the user has at least read privileges for.

The default grants of the above mentioned roles are as follows:

1. The Administrator role privileges

Role	**System Privilege**
scheduler-administrator	Application.Create
scheduler-administrator	Application.Delete
scheduler-administrator	Application.Modify
scheduler-administrator	Application.View
scheduler-administrator	EventDefinition.Clear
scheduler-administrator	EventDefinition.Create
scheduler-administrator	EventDefinition.Delete
scheduler-administrator	EventDefinition.Modify
scheduler-administrator	EventDefinition.Raise
scheduler-administrator	EventDefinition.View
scheduler-administrator	Format.Create
scheduler-administrator	Format.Delete
scheduler-administrator	Format.Modify
scheduler-administrator	Format.View
scheduler-administrator	Job.Delete
scheduler-administrator	Job.Modify
scheduler-administrator	Job.View
scheduler-administrator	JobDefinition.Create
scheduler-administrator	JobDefinition.Delete
scheduler-administrator	JobDefinition.DeleteJob
scheduler-administrator	JobDefinition.Modify
scheduler-administrator	JobDefinition.SubmitJob
scheduler-administrator	JobDefinition.View
scheduler-administrator	JobDefinitionType.Create
scheduler-administrator	JobDefinitionType.Delete
scheduler-administrator	JobDefinitionType.Modify
scheduler-administrator	JobDefinitionType.View
scheduler-administrator	JobFile.View
scheduler-administrator	JobLock.Create
scheduler-administrator	JobLock.Delete
scheduler-administrator	JobLock.Modify
scheduler-administrator	JobLock.View
scheduler-administrator	ProcessServer.Create
scheduler-administrator	ProcessServer.Delete
scheduler-administrator	ProcessServer.Modify
scheduler-administrator	ProcessServer.View

scheduler-administrator	Queue.Create
scheduler-administrator	Queue.Delete
scheduler-administrator	Queue.DeleteJobIn
scheduler-administrator	Queue.Modify
scheduler-administrator	Queue.SubmitJobIn
scheduler-administrator	Queue.View
scheduler-administrator	RegistryEntry.Create
scheduler-administrator	Resource.Create
scheduler-administrator	Resource.Delete
scheduler-administrator	Resource.Modify
scheduler-administrator	Resource.View
scheduler-administrator	SAPSystem.Create
scheduler-administrator	SAPSystem.Delete
scheduler-administrator	SAPSystem.Modify
scheduler-administrator	SAPSystem.View
scheduler-administrator	Service.Create
scheduler-administrator	Service.Delete
scheduler-administrator	Service.Modify
scheduler-administrator	Service.View
scheduler-administrator	Subject.CanGrant
scheduler-administrator	Subject.Create
scheduler-administrator	Subject.Delete
scheduler-administrator	Subject.Modify
scheduler-administrator	Subject.View
scheduler-administrator	SubmitFrame.Create
scheduler-administrator	SubmitFrame.Delete
scheduler-administrator	SubmitFrame.Modify
scheduler-administrator	SubmitFrame.View
scheduler-administrator	TimeWindow.Create
scheduler-administrator	TimeWindow.Delete
scheduler-administrator	TimeWindow.Modify
scheduler-administrator	TimeWindow.View

2. Secondary role privileges

Role	**System Privilege**
scheduler-job-administrator	EventDefinition.Create
scheduler-job-administrator	EventDefinition.Delete
scheduler-job-administrator	EventDefinition.Modify
scheduler-job-administrator	Job.Modify
scheduler-job-administrator	JobDefinition.Create

scheduler-job-administrator	JobDefinition.Delete
scheduler-job-administrator	JobDefinition.DeleteJob
scheduler-job-administrator	JobDefinition.Modify
scheduler-job-administrator	JobDefinition.SubmitJob
scheduler-job-administrator	Queue.DeleteJobIn
scheduler-job-administrator	Queue.SubmitJobIn
scheduler-event-operator	EventDefinition.Clear
scheduler-event-operator	EventDefinition.Raise

3. The generic role privileges

Role	**System Privilege**
scheduler-viewer	Application.View
scheduler-viewer	EventDefinition.View
scheduler-viewer	Format.View
scheduler-viewer	Job.View
scheduler-viewer	JobDefinition.View
scheduler-viewer	JobDefinitionType.View
scheduler-viewer	JobFile.View
scheduler-viewer	JobLock.View
scheduler-viewer	ProcessServer.View
scheduler-viewer	Queue.View
scheduler-viewer	Resource.View
scheduler-viewer	SAPSystem.View
scheduler-viewer	Service.View
scheduler-viewer	Subject.View
scheduler-viewer	SubmitFrame.View
scheduler-viewer	TimeWindow.View

Object Privileges

Use

Object privileges are used to grant specific users access to specific objects. Unlike system privileges, object privileges are only valid for one object. Object privileges require the *Security.Module* which is specified in a license key. You know you have object privileges when you see a *Security* tab on repository objects, like job definitions, for example.

The following basic object privileges are available for most objects:

- View
- Edit
- Delete

Some Objects have additional privileges, for example, two more object-specific object privileges exist for job definitions:

- Submit
- View Jobs

Object privileges can be granted with the *Access* or *Admin* option; the *Admin* option allows the grantee to grant the privilege on to other users. Object privileges cannot be granted directly, you grant ranks of privileges. For example, the *Edit* rank contains both *View* and *Edit* privileges, this prevents human error, as you need to see an object before you can edit it.

See Also

Granting and Revoking Object Privileges

Granting and Revoking System Privileges

System Privileges

Granting and Revoking System Privileges

Use

System privileges can be granted to a custom role for all objects system-wide, in a partition or with an isolation group. Custom roles are created in the authentication system, they are created in SAP CPS as soon as a user with that role logs on.

You cannot modify any of the built-in roles:

- `scheduler-administrator` - can perform all actions.
- `scheduler-event-operator` - can raise and clear events.
- `scheduler-job-administrator` - can create/edit/delete event definitions, job definitions, job chains. Can modify jobs.
- `scheduler-user` - has access to SAP CPS only, cannot see any objects.
- `scheduler-viewer` - read only access to all objects.
- `scheduler-isolation-administrator` - can create/edit/delete isolation groups and add users to these.
- `scheduler-screen-reader` - indicates that you are using a screen reader (not displayed when running in SAP NetWeaver).

Procedure

Granting privileges to a custom role

1. Navigate to *Security* > *Roles*.
2. Choose *Edit* from the context menu of an editable role. Editable roles have a description: *Created automatically on first login*.
3. On the *Assign Privileges* tab, choose an Object definition and then *Next*.

4. Choose the desired range of the privileges.
5. Choose a *Rank* with the desired privileges. *Admin* privileges allow the user to perform the action and to grant the privilege to others as well. *Access* privileges allow the user to perform the actions.

Revoking privileges from a custom role

1. Navigate to *Security* > *Roles*.
2. Choose *Edit* from the context menu of an editable role.Editable roles have a description: *Created automatically on first login*.
3. On the *Assign Privileges* tab, choose an Object definition and then *Next*.
4. Choose the desired range of the privileges.
5. Choose a *Rank* with the desired privileges. *Admin* privileges allow the user to perform the action and to grant the privilege to others as well. *Access* privileges allow the user to perform the actions.

See Also

- Granting or Revoking Object Privileges

Granting and Revoking Object Privileges

Use

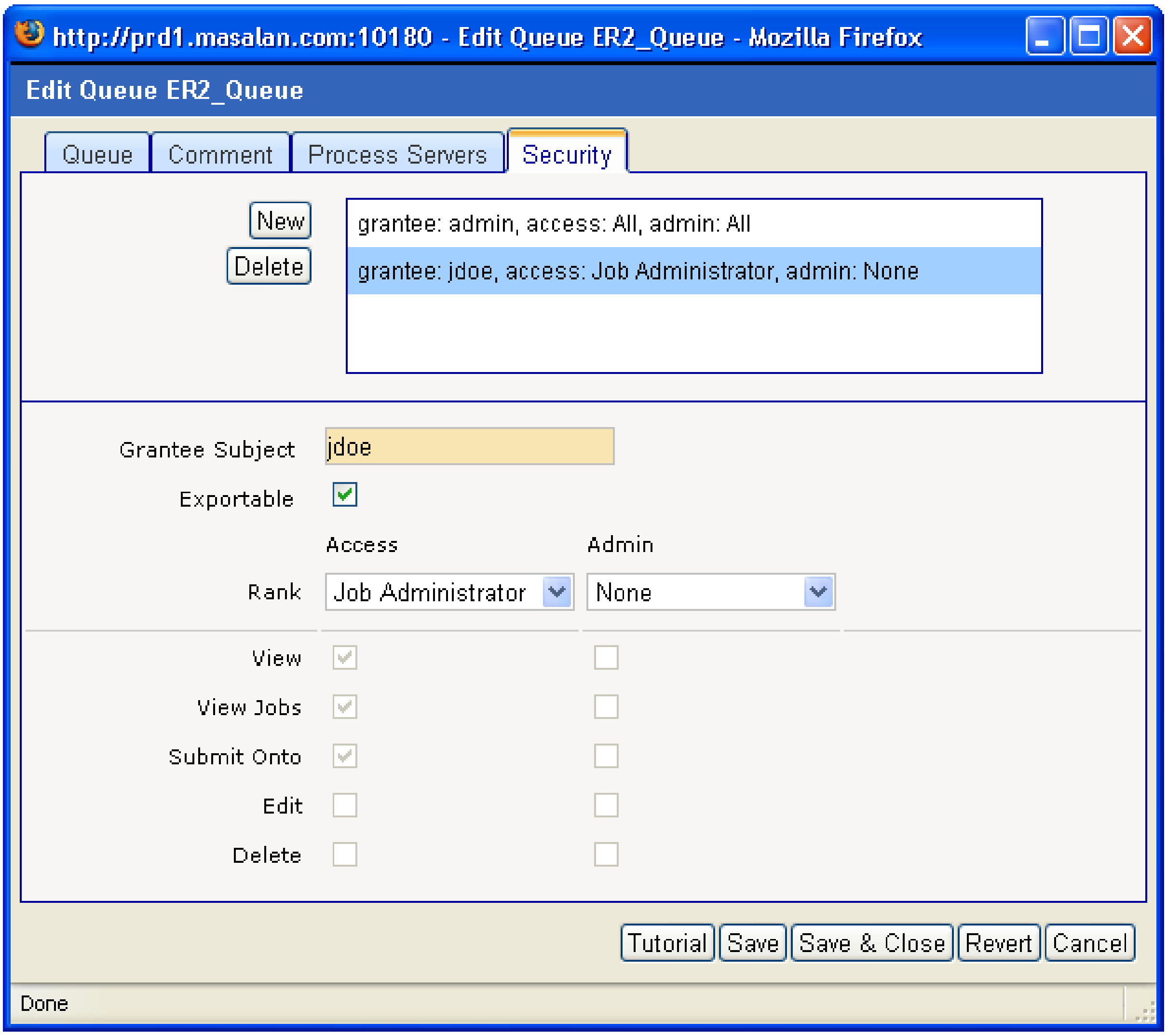
Object privileges can be granted/revoked to/from a custom role or user for a specific object. You cannot modify any of the built-in roles:

- `scheduler-administrator` - can perform all actions.
- `scheduler-event-operator` - can raise and clear events.
- `scheduler-job-administrator` - can create/edit/delete event definitions, job definitions, job chains. Can modify jobs.
- `scheduler-user` - has access to SAP CPS only, cannot see any objects.
- `scheduler-viewer` - read only access to all objects.
- `scheduler-isolation-administrator` - can create/edit/delete isolation groups and add users to these.
- `scheduler-screen-reader` - indicates that you are using a screen reader (not displayed when running in SAP NetWeaver).

Object privileges cannot be granted directly, you grant ranks of privileges. For example, the *Job Administrator* rank contains *View* and *View Jobs* and *Submit Onto* privileges, this prevents human error, as you need to be able to see the queue, the jobs in that queue and you need to be able to submit a job into the queue to be able to be able to submit a job into the queue and monitor its status. Furthermore, privileges can be granted as *Access* and *Admin*, when you grant a privilege as *Admin*, the grantee can grant the privilege to other users.

Exportable privileges will be exported together with the object, the grantee has to exist in the system where you import the object.

The figure below illustrates the privileges of user *jdoe* on the queue *ER2_Queue*.



grantee: admin, access: All, admin: All

grantee: jdoe, access: Job Administrator, admin: None

Grantee Subject: jdoe

Exportable: ☑

	Access	Admin
Rank	Job Administrator	None
View	☑	□
View Jobs	☑	□
Submit Onto	☑	□
Edit	□	□
Delete	□	□

Procedure

Granting object privileges

1. Navigate to the object you want to change the security for.

2. Choose *Edit Security* from the context menu.

3. On the *Security* tab, choose *New* and select the grantee.

4. Choose a *Rank* with the desired privileges. *Admin* privileges allow the user to perform the action and to grant the privilege to others as well. *Access* privileges allow the user to perform the actions.

Revoking object privileges

1. Navigate to the object you want to change the security for.

2. Choose *Edit Security* from the context menu.

3. On the *Security* tab, choose the grantee.

4. Choose a *Rank* with the desired privileges, if you want to revoke all privileges for this object from the grantee, choose *Delete*. *Admin* privileges allow the user to perform the action and to grant the privilege to others as well. *Access* privileges allow the user to perform the actions.

Example

Granting *Submit* privileges on the job definition *ProcessPartnerFile* to user *jdoe*.

```
{
//Get the job definition and user
JobDefinition jd = jcsSession.getJobDefinitionByName("ProcessPartnerFile");
Subject usr = jcsSession.getSubjectByTypeName(SubjectType.User,"jdoe");
//Create an empty grant
SubjectObjectPrivilegeGrant grant = jd.createSubjectObjectPrivilegeGrant();
//Set the grantee and the rank
grant.setGranteeSubject(usr);
grant.setGrantedRank(grant.getAllRanks()[3]);
//Save the changes to the database
jcsSession.persist();
}
```

See Also

- Granting and Revoking System Privileges

Object Privileges

Use

Object privileges are used to grant specific users access to specific objects. Unlike system privileges, object privileges are only valid for one object. Object privileges require the *Security.Module* which is specified in a license key. You know you have object privileges when you see a *Security* tab on repository objects, like job definitions, for example.

The following basic object privileges are available for most objects:

- View
- Edit
- Delete

Some Objects have additional privileges, for example, two more object-specific object privileges exist for job definitions:

- Submit
- View Jobs

Object privileges can be granted with the *Access* or *Admin* option; the *Admin* option allows the grantee to grant the privilege on to other users. Object privileges cannot be granted directly, you grant ranks of privileges. For example, the *Edit* rank contains both *View* and *Edit* privileges, this prevents human error, as you need to see an object before you can edit it.

See Also

Granting and Revoking Object Privileges

Granting and Revoking System Privileges

System Privileges

Granting and Revoking System Privileges

Use

System privileges can be granted to a custom role for all objects system-wide, in a partition or with an isolation group. Custom roles are created in the authentication system, they are created in SAP CPS as soon as a user with that role logs on.

You cannot modify any of the built-in roles:

- `scheduler-administrator` - can perform all actions.
- `scheduler-event-operator` - can raise and clear events.
- `scheduler-job-administrator` - can create/edit/delete event definitions, job definitions, job chains. Can modify jobs.
- `scheduler-user` - has access to SAP CPS only, cannot see any objects.
- `scheduler-viewer` - read only access to all objects.
- `scheduler-isolation-administrator` - can create/edit/delete isolation groups and add users to these.
- `scheduler-screen-reader` - indicates that you are using a screen reader (not displayed when running in SAP NetWeaver).

Procedure

Granting privileges to a custom role

1. Navigate to *Security* > *Roles*.
2. Choose *Edit* from the context menu of an editable role. Editable roles have a description: *Created automatically on first login*.
3. On the *Assign Privileges* tab, choose an Object definition and then *Next*.
4. Choose the desired range of the privileges.
5. Choose a *Rank* with the desired privileges. *Admin* privileges allow the user to perform the action and to grant the privilege to others as well. *Access* privileges allow the user to perform the actions.

Revoking privileges from a custom role

1. Navigate to *Security* > *Roles*.
2. Choose *Edit* from the context menu of an editable role.Editable roles have a description: *Created automatically on first login*.
3. On the *Assign Privileges* tab, choose an Object definition and then *Next*.
4. Choose the desired range of the privileges.

5. Choose a *Rank* with the desired privileges. *Admin* privileges allow the user to perform the action and to grant the privilege to others as well. *Access* privileges allow the user to perform the actions.

See Also

- Granting or Revoking Object Privileges

Granting and Revoking Object Privileges

Use

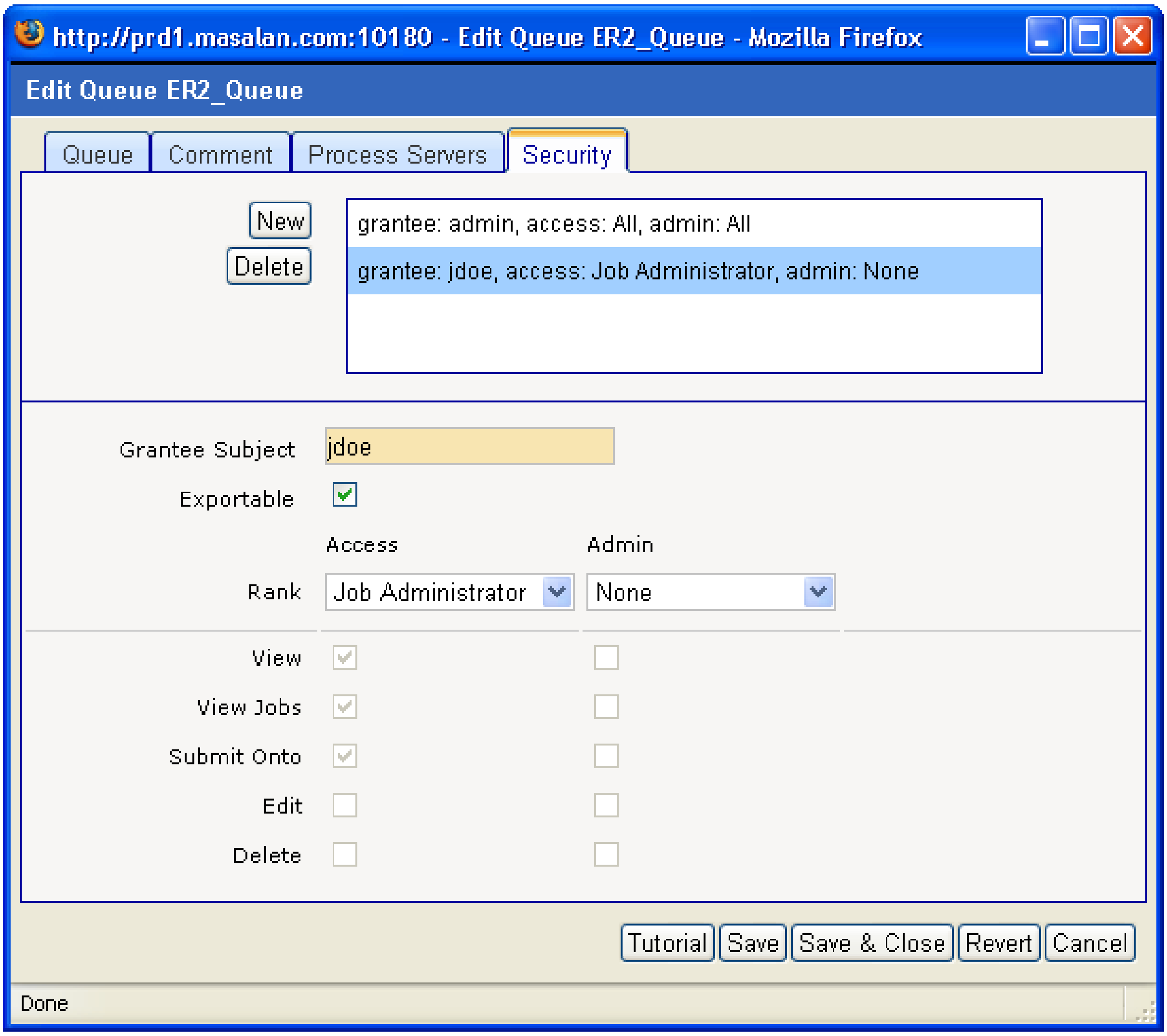
Object privileges can be granted/revoked to/from a custom role or user for a specific object. You cannot modify any of the built-in roles:

- `scheduler-administrator` - can perform all actions.
- `scheduler-event-operator` - can raise and clear events.
- `scheduler-job-administrator` - can create/edit/delete event definitions, job definitions, job chains. Can modify jobs.
- `scheduler-user` - has access to SAP CPS only, cannot see any objects.
- `scheduler-viewer` - read only access to all objects.
- `scheduler-isolation-administrator` - can create/edit/delete isolation groups and add users to these.
- `scheduler-screen-reader` - indicates that you are using a screen reader (not displayed when running in SAP NetWeaver).

Object privileges cannot be granted directly, you grant ranks of privileges. For example, the *Job Administrator* rank contains *View* and *View Jobs* and *Submit Onto* privileges, this prevents human error, as you need to be able to see the queue, the jobs in that queue and you need to be able to submit a job into the queue to be able to be able to submit a job into the queue and monitor its status. Furthermore, privileges can be granted as *Access* and *Admin*, when you grant a privilege as *Admin*, the grantee can grant the privilege to other users.

Exportable privileges will be exported together with the object, the grantee has to exist in the system where you import the object.

The figure below illustrates the privileges of user *jdoe* on the queue *ER2_Queue*.



grantee: admin, access: All, admin: All

grantee: jdoe, access: Job Administrator, admin: None

Grantee Subject: jdoe

Exportable: ☑

	Access	Admin
Rank	Job Administrator	None
View	☑	☐
View Jobs	☑	☐
Submit Onto	☑	☐
Edit	☐	☐
Delete	☐	☐

Procedure

Granting object privileges

1. Navigate to the object you want to change the security for.

2. Choose *Edit Security* from the context menu.

3. On the *Security* tab, choose *New* and select the grantee.

4. Choose a *Rank* with the desired privileges. *Admin* privileges allow the user to perform the action and to grant the privilege to others as well. *Access* privileges allow the user to perform the actions.

Revoking object privileges

1. Navigate to the object you want to change the security for.

2. Choose *Edit Security* from the context menu.

3. On the *Security* tab, choose the grantee.

4. Choose a *Rank* with the desired privileges, if you want to revoke all privileges for this object from the grantee, choose *Delete*. *Admin* privileges allow the user to perform the action and to grant the privilege to others as well. *Access* privileges allow the user to perform the actions.

Example

Granting *Submit* privileges on the job definition *ProcessPartnerFile* to user *jdoe*.

```
{
//Get the job definition and user
JobDefinition jd = jcsSession.getJobDefinitionByName("ProcessPartnerFile");
Subject usr = jcsSession.getSubjectByTypeName(SubjectType.User,"jdoe");
//Create an empty grant
SubjectObjectPrivilegeGrant grant = jd.createSubjectObjectPrivilegeGrant();
//Set the grantee and the rank
grant.setGranteeSubject(usr);
grant.setGrantedRank(grant.getAllRanks()[3]);
//Save the changes to the database
jcsSession.persist();
}
```

See Also

- Granting and Revoking System Privileges

Privileges Required to use Objects

Use

The following sections cover the privileges that are required to use SAP CPS objects.

- Privileges Required to use Ad Hoc Alert Sources
- Privileges Required to use AlertSourceActions
- Privileges Required to use Applications
- Privileges Required to use Audit Rules
- Privileges Required to use the Audit Trail
- Privileges Required to use CounterCategories
- Privileges Required to use Credential Protocols
- Privileges Required to use Credentials
- Privileges Required to use Dashboards
- Privileges Required to use DatumDefinitions
- Privileges Required to use Alert Escalations
- Privileges Required to use Events and Event Definitions
- Privileges Required to use ExportRuleSets
- Privileges Required to use Export Rule Sets

- Privileges Required to use Formats
- Privileges Required to use Email Alert Gateways
- Privileges Required to use ImportRuleDefinitions
- Privileges Required to use ImportRuleSets
- Privileges Required to use ImportSources
- Privileges Required to use Import Rule Sets
- Privileges Required to use Isolation Groups
- Privileges Required to use Job Chains
- Privileges Required to use Job Definition Alert Sources
- Privileges Required to use Job Definition Types
- Privileges Required to use Job Definitions
- Privileges Required to use Jobs
- Privileges Required to use Libraries
- Privileges Required to use Locks
- Privileges Required to use Monitor Nodes
- Privileges Required to use Operator Messages
- Privileges Required to use Partitions
- Privileges Required to use Period Functions
- Privileges Required to use Process Server Alert Sources
- Privileges Required to use Process Servers
- Privileges Required to use QueryFilters
- Privileges Required to use Queue Alert Source
- Privileges Required to use Queues
- Privileges Required to use Registry Entries
- Privileges Required to use Reports
- Privileges Required to use Resources
- Privileges Required to use Roles
- Privileges Required to use SAP Systems
- Privileges Required to use Services
- Privileges Required to use the Shell

- Privileges Required to use StatisticCagtegories
- Privileges Required to use SubjectIsolationGroups
- Privileges Required to use SubjectRoleGrantExpanded
- Privileges Required to use Subjects
- Privileges Required to use Submit Frames
- Privileges Required to use Table Definitions
- Privileges Required to use Tables
- Privileges Required to use Time Windows
- Privileges Required to use Time Zones
- Privileges Required to use Triggers
- Privileges Required to use Users

Privileges Required to use Ad Hoc Alert Sources

Use

To use ad hoc alert sources, you need either:

- `scheduler-administrator` role
- system, partition or isolation group-level permissions
- object-specific privileges

Creating Ad Hoc Alert Sources

You need the either of the following privilege ranks to be able to create ad hoc alert sources:

- *Create* - allows you to create an ad hoc alert source on the level the privilege was granted (system, partition, isolation-group), you have no further privileges through this rank, you automatically get privileges on ad hoc alert sources you create.
- *Edit* - allows you to create, view, and edit all ad hoc alert sources on the level the privilege was granted (system, partition, isolation-group)
- *Delete* - allows you to create, view, and delete all ad hoc alert source on the level the privilege was granted (system, partition, isolation-group)
- *All* - full control over all ad hoc alert sources on the level the privilege was granted (system, partition, isolation-group)

You always need *View* privileges on the partition in which you want to create the ad hoc alert source.

Creating Alert Source Actions

You need the either of the following privilege ranks to be able to create ad hoc alert sources:

- *Create* - allows you to create an alert source action on the level the privilege was granted (system, partition, isolation-group), you have no further privileges through this rank, you automatically get privileges on ad hoc alert sources you create.
- *Edit* - allows you to create, view, and edit all alert source actions on the level the privilege was granted (system, partition, isolation-group)
- *Delete* - allows you to create, view, and delete all alert source actions on the level the privilege was granted (system, partition, isolation-group)
- *All* - full control over all alert source actions on the level the privilege was granted (system, partition, isolation-group)

Editing Ad Hoc Alert Sources

To successfully edit a ad hoc alert source, you have to have either of the following privileges:

- *Edit* - privilege rank on the ad hoc alert source, or on ad hoc alert sources in its partition, isolation group or system-wide
- *Delete* - privilege rank on the ad hoc alert source, or on ad hoc alert sources in its partition, isolation group or system-wide
- *All* - privilege rank on the ad hoc alert source, or on ad hoc alert sources in its partition, isolation group or system-wide

Ad hoc alert sources can reference the following objects, you need at least *View* privileges on these objects as well as their partitions when you want to edit an ad hoc alert source that references them:

- *Application*
- *Alert Escalation*

You always need *View* privileges on the partition of the ad hoc alert source.

Editing Alert Source Actions

To successfully edit a ad hoc alert source, you have to have either of the following privileges:

- *Edit* - privilege rank on the alert source action, or on alert source actions in its partition, isolation group or system-wide
- *Delete* - privilege rank on the alert source action, or on alert source actions in its partition, isolation group or system-wide
- *All* - privilege rank on the alert source action, or on alert source actions in its partition, isolation group or system-wide

See Also

- Privileges Required to use Applications
- Privileges Required to use Alert Escalations

Privileges Required to use AlertSourceActions

Use

To use AlertSourceActions you need either:

- `scheduler-administrator` role
- system, partition or isolation group-level permissions on the alert source and system or isolation group-level permissions on AlertSourceAction
- object-specific privileges on the alert source and system or isolation group-level permissions on AlertSource-Action

Note

AlertSourceActions are properties of alert sources and as such you require permissions on the mother-object as well, in this case the alert source.

Creating AlertSourceActions

You need either of the following privilege ranks to be able to create AlertSourceActions:

- *Create* - allows you to create AlertSourceActions on the level the privilege was granted (system, partition, isolation-group), you have no further privileges through this rank, you automatically get privileges on AlertSourceActions you create.
- *Edit* - allows you to create, view and edit all AlertSourceActions on the level the privilege was granted (system, partition, isolation-group, object)
- *Delete* - allows you to create, view, manage and delete all AlertSourceActions and AlertSourceActions on the level the privilege was granted (system, partition, isolation-group, object)
- *All* - full control over all AlertSourceActions and AlertSourceActions on the level the privilege was granted (system, partition, isolation-group, object)

Editing AlertSourceActions

To successfully edit an AlertSourceAction, you have to have at least one of the following privileges:

- *Edit* - privilege rank on the AlertSourceAction, or on AlertSourceActions in its partition, isolation group or system-wide
- *All* - privilege rank on the AlertSourceAction, or on AlertSourceActions in its partition, isolation group or system-wide

See Also

- Privileges Required to use Applications
- Privileges Required to use Process Servers
- Privileges Required to use AdHoc Alert Sources
- Privileges Required to use Job Definition Alert Sources
- Privileges Required to use Process Server Alert Sources

- Privileges Required to use Queue Alert Sources

Privileges Required to use Applications

To use applications, you need either:

- `scheduler-administrator` role
- system, partition or isolation group-level permissions
- object-specific privileges

Creating and Editing Applications

You need the following privilege ranks to be able to create applications:

- *Create* - allows you to create applications on the level the privilege was granted (system, partition, isolation-group), you have no further privileges through this rank, you automatically get privileges on applications you create.
- *Edit* - allows you to create, view and edit all applications on the level the privilege was granted (system, partition, isolation-group)
- *Delete* - allows you to create, view, and delete all applications on the level the privilege was granted (system, partition, isolation-group, object)
- *All* - full control over all applications on the level the privilege was granted (system, partition, isolation-group)

You always need *View* privileges on the partition in which you want to create the application.

Editing Applications

You need the following privilege ranks to be able to edit applications:

- "Edit- privilege rank on the application, or on applications in its partition, isolation group or system-wide
- "All- privilege rank on the application, or on applications in its partition, isolation group or system-wide

You always need *View* privileges on the partition of the application.

Privileges Required to use Audit Rules

Use

To use audit rules, you need either:

- `scheduler-administrator` role
- system, partition or isolation group-level permissions
- object-specific privileges

Creating Audit Rules

You need the following privilege ranks to be able to create audit rules:

- *Create* - allows you to create audit rules on the level the privilege was granted (system, partition, isolation-group), you have no further privileges through this rank, you automatically get privileges on audit rules you create.

- *Edit* - allows you to create, view and edit all audit rules on the level the privilege was granted (system, partition, isolation-group, object)
- *Delete* - allows you to create, view, and delete all audit rules on the level the privilege was granted (system, partition, isolation-group, object)
- *All* - full control over all audit rules on the level the privilege was granted (system, partition, isolation-group, object)

Editing Audit Rules

- *Edit* - privilege rank on the audit rule, or on audit rules in its partition, isolation group or system-wide
- *All* - privilege rank on the audit rule, or on audit rules in its partition, isolation group or system-wide

Audit rules can reference the following objects, you need at least *View* privileges on these objects and their partitions when you want to edit an audit rule that references them:

- *Application*

Deleting Audit Rules

- *Delete*
- *All*

Privileges Required to use the Audit Trail

Use

To view the audit trail, you need either:

- `scheduler-administrator` role
- system or isolation group-level permissions

Viewing Audit Trail AuditObjects

- *View* - allows you to view AuditObjects on the level the privilege was granted (system, isolation-group), you have no further privileges through this rank.
- *Edit* - allows you to view and restore all AuditObjects on the level the privilege was granted (system, isolation-group)
- *Delete* - allows you to view, restore and delete any AuditObject on the level the privilege was granted (system, isolation-group)
- *All* - full control over all AuditObjects on the level the privilege was granted (system, isolation-group)

Note

A user with the *View* privilege on AuditObject does not require - privilege rank on the audited object to see it in the audit trail., or on audited object to see it in the audit trail.

Restoring Objects

- *Edit* - allows you to create, view and restore all AuditObjects on the level the privilege was granted (system, isolation-group)

- *Delete* - allows you to view, restore and delete any AuditObjects on the level the privilege was granted (system, isolation-group)
- *All* - full control over all AuditObjects on the level the privilege was granted (system, isolation-group)

See Also

- Privileges Required to use Isolation Groups
- Privileges Required to use Partitions

Privileges Required to use CounterCategories

Use

CounterCategories are used to calculate job statistics.

To use CounterCategories, you need either:

- `scheduler-administrator` role
- system, partition or isolation group-level permissions
- object-specific privileges

Creating CounterCategories

You need the either of the following privilege ranks to be able to create CounterCategories:

- *Create* - allows you to create an CounterCategory on the level the privilege was granted (system, partition, isolation-group), you have no further privileges through this rank, you automatically get privileges on CounterCategories you create.
- *Edit* - allows you to create, view, and edit all CounterCategories on the level the privilege was granted (system, partition, isolation-group)
- *Delete* - allows you to create, view, and delete all CounterCategory on the level the privilege was granted (system, partition, isolation-group)
- *All* - full control over all CounterCategories on the level the privilege was granted (system, partition, isolation-group)

Editing CounterCategories

To successfully edit a CounterCategory, you have to have either of the following privileges:

- *Edit* - privilege rank on the CounterCategory, or on CounterCategories in its partition, isolation group or system-wide
- *Delete* - privilege rank on the CounterCategory, or on CounterCategory in its partition, isolation group or system-wide
- *All* - privilege rank on the CounterCategory, or on CounterCategories in its partition, isolation group or system-wide

See Also

StatisticCategory

Privileges Required to use Credential Protocols

Use

To use credential protocols, you need either:

- `scheduler-administrator` role
- system, partition or isolation group-level permissions
- object-specific privileges

Using Credential Protocols

You need the following privilege ranks to be able to use credential protocols in job definitions:

- *View* - allows you to view the credential protocols and use it in a job definition.

Creating Credential Protocols

- *Create* - allows you to create credential protocols on the level the privilege was granted (system, partition, isolation-group), you have no further privileges through this rank, you automatically get privileges on audit rules you create.
- *Edit* - allows you to create, view and edit all credential protocols on the level the privilege was granted (system, partition, isolation-group, object)
- *Delete* - allows you to create, view, and delete all credential protocols on the level the privilege was granted (system, partition, isolation-group, object)
- *All* - full control over all credential protocols on the level the privilege was granted (system, partition, isolation-group, object)

You always need *View* privileges on the partition in which you want to create the credential protocol.

Note

You can create credential protocols, however, you will only be able to use them with specially crafted job definition types.

Editing Credential Protocols

- *Edit* - privilege rank on the credential protocol, or on credential protocols in its partition, isolation group or system-wide
- *All* - privilege rank on the credential protocol, or on credential protocols in its partition, isolation group or system-wide

You always need *View* privileges on the partition of the credential protocol.

Credential protocols can reference the following objects, you need at least *View* privileges on these objects and their partitions when you want to edit a credential protocols that references them:

- *Application*

Deleting Credential Protocols

- *Delete* privilege on the credential protocol, or on credential protocols in its partition, isolation group or system-wide
- *All* credential protocol, or on credential protocols in its partition, isolation group or system-wide

You always need *View* privileges on the partition of the credential protocol.

See Also

- Privileges Required to use Applications

Privileges Required to use Credentials

Use

To use credentials, you need either:

- `scheduler-administrator` role
- system, partition or isolation group-level permissions
- object-specific privileges

Using Credentials

You need the following privilege ranks to be able to use credentials in job definitions:

- *View* - allows you to view the credential and use it in a job definition.

You always need *View* privileges on the partition of the credential protocol.

Creating Credentials

You need the following privilege ranks to be able to create credentials:

- *Create* - allows you to create credential on the level the privilege was granted (system, partition, isolation-group), you have no further privileges through this rank, you automatically get privileges on audit rules you create.
- *Edit* - allows you to create, view and edit all credentials on the level the privilege was granted (system, partition, isolation-group, object)
- *All* - full control over all credentials on the level the privilege was granted (system, partition, isolation-group, object)

You always need *View* privileges on the partition in which you want to create the credential.

Editing Credentials

- *Edit* - allows you to create, view and edit all credentials on the level the privilege was granted (system, partition, isolation-group, object)
- *All* - full control over all credentials on the level the privilege was granted (system, partition, isolation-group, object)

You always need *View* privileges on the partition of the credential.

Credentials can reference the following objects, you need at least *View* privileges on these objects and their partitions when you want to edit a credential that references them:

- *Application*
- *Credential Protocol*

Deleting Credentials

- *Delete* - allows you to view and delete all credentials on the level the privilege was granted (system, partition, isolation-group, object)
- *All* - full control over all credentials on the level the privilege was granted (system, partition, isolation-group, object)

You always need *View* privileges on the partition of the credential.

See Also

- Privileges Required to use Applications
- Privileges Required to use Credential Protocols

Privileges Required to use Dashboards

Use

To use dashboards, you need either:

- `scheduler-administrator` role
- `scheduler-job-administrator` role

Creating Dashboards

You need one of the following privilege ranks to be able to create dashboards:

- *Create* - allows you to create dashboards
- *Edit* - allows you to create, view, and edit all dashboards
- *Delete* - allows you to create, view, and delete all dashboards
- *All* - full control over all dashboards

Editing Dashboards

To successfully edit a dashboard, you have to have one the following privileges:

- *Edit* - system-wide privilege rank for the dashboard
- *Delete* - system-wide privilege rank for the dashboard
- *All* - system-wide privilege rank for the dashboard

Privileges Required to use DatumDefinitions

Use

To use DatumDefinitions you need either:

- `scheduler-administrator` role
- system, partition or isolation group-level permissions on the alert source and on DatumDefinition
- object-specific privileges on the object and system, partition or isolation group-level permissions on DatumDefinition

> **Note**
>
> DatumDefinitions are properties of objects and as such you require permissions on the mother-object as well.

Creating DatumDefinitions

You need either of the following privilege ranks to be able to create DatumDefinitions:

- *Create* - allows you to create DatumDefinitions on the level the privilege was granted (system, partition, isolation-group), you have no further privileges through this rank, you automatically get privileges on DatumDefinitions you create.
- *Edit* - allows you to create, view and edit all DatumDefinitions on the level the privilege was granted (system, partition, isolation-group, object)
- *Delete* - allows you to create, view, manage and delete all DatumDefinitions and DatumDefinitions on the level the privilege was granted (system, partition, isolation-group, object)
- *All* - full control over all DatumDefinitions and DatumDefinitions on the level the privilege was granted (system, partition, isolation-group, object)

Editing DatumDefinitions

To successfully edit an DatumDefinition, you have to have at least one of the following privileges:

- *Edit* - privilege rank on the DatumDefinition, or on DatumDefinitions in its partition, isolation group or system-wide
- *All* - privilege rank on the DatumDefinition, or on DatumDefinitions in its partition, isolation group or system-wide

Privileges Required to use Alert Escalations

Use

To use alert escalations, you need either:

- `scheduler-administrator` role
- system, partition or isolation group-level permissions
- object-specific privileges

Creating Alert Escalations

You need one of the following privilege ranks to be able to create escalations:

- *Create* - allows you to create alert escalations on the level the privilege was granted (system, partition, isolation-group), you have no further privileges through this rank, you automatically get privileges on alert escalations you create.
- *Edit* - allows you to create, view and edit all alert escalations on the level the privilege was granted (system, partition, isolation-group, object)
- *All* - full control over all alert escalations on the level the privilege was granted (system, partition, isolation-group, object)

You always need *View* privileges on the partition in which you want to create the alert escalation.

Editing Escalations

To successfully edit alert escalations, you have to have one the following privileges:

- *Edit* - allows you to create, view and edit all alert escalations on the level the privilege was granted (system, partition, isolation-group, object)
- *All* - full control over all alert escalations on the level the privilege was granted (system, partition, isolation-group, object)

Alert escalations can reference the following objects, you need at least *View* privileges on these objects and their partitions when you want to edit an alert escalation that references them:

- *Application*

You always need *View* privileges on the partition of the alert escalation.

See Also

- Privileges Required to use Applications

Privileges Required to use Events and Event Definitions

Use

To use event definitions and their events, you need either:

- `scheduler-administrator` role
- `scheduler-event-operator` role
- system, partition or isolation group-level permissions
- object-specific privileges

Creating Event Definitions

You need either of the following privilege ranks to be able to create event definitions:

- *Create* - allows you to create events on the level the privilege was granted (system, partition, isolation-group), you have no further privileges through this rank, you automatically get privileges on events you create.

- *Edit* - allows you to create, view and edit all events on the level the privilege was granted (system, partition, isolation-group, object)
- *Delete* - allows you to create, view, manage and delete all event definitions and events on the level the privilege was granted (system, partition, isolation-group, object)
- *All* - full control over all event definitions and events on the level the privilege was granted (system, partition, isolation-group, object)

You always need *View* privileges on the partition in which you want to create the event definition.

Editing Event Definitions

To successfully edit an event definition, you have to have at least one of the following privileges:

- *Edit* - privilege rank on the event definition, or on event definitions in its partition, isolation group or system-wide
- *All* - privilege rank on the event definition, or on event definitions in its partition, isolation group or system-wide

You always need *View* privileges on the partition of the event definition.

Event definitions can reference the following objects, you need at least *View* privileges on these objects and their partitions when you want to edit an event definition that references them:

- *Application*
- *Process Server* - file events only

Editing Events

- *Clear* - privilege rank on the event definition, it allows you to clear events on the level the privilege was granted (system, partition, isolation-group, object), or on event definition, it allows you to clear events on the level the privilege was granted (system, partition, isolation-group, object) in its partition, isolation group or system-wide
- *Raise* - privilege rank on the event definition, it allows you to raise events on the level the privilege was granted (system, partition, isolation-group, object), or on event definition, it allows you to raise events on the level the privilege was granted (system, partition, isolation-group, object) in its partition, isolation group or system-wide
- *Manage Events* - privilege rank on the event definition, it allows you to raise and clear events on the level the privilege was granted (system, partition, isolation-group, object), or on event definition, it allows you to raise and clear events on the level the privilege was granted (system, partition, isolation-group, object) in its partition, isolation group or system-wide
- *Edit* - privilege rank on the event definition, or on event definitions in its partition, isolation group or system-wide
- *All* - privilege rank on the event definition, or on event definitions in its partition, isolation group or system-wide

You always need *View* privileges on the partition of the event definition.

See Also

- Privileges Required to use Applications

- Privileges Required to use Process Servers

Privileges Required to use ExportRuleSets

Use

To use ExportRuleSets you need either:

- `scheduler-administrator` role
- system, partition or isolation group-level permissions
- object-specific privileges

Creating ExportRuleSets

You need either of the following privilege ranks to be able to create ExportRuleSets:

- *Create* - allows you to create ExportRuleSets on the level the privilege was granted (system, partition, isolation-group), you have no further privileges through this rank, you automatically get privileges on ExportRuleSets you create.
- *Edit* - allows you to create, view and edit all ExportRuleSets on the level the privilege was granted (system, partition, isolation-group, object)
- *Delete* - allows you to create, view, and delete all ExportRuleSets on the level the privilege was granted (system, partition, isolation-group, object)
- *All* - full control over all ExportRuleSets on the level the privilege was granted (system, partition, isolation-group, object)

You always need *View* privileges on the partition in which you want to create the ExportRuleSet.

Editing ExportRuleSets

To successfully edit an ExportRuleSet, you have to have at least one of the following privileges:

- *Edit* - privilege rank on the ExportRuleSet, or on ExportRuleSets in its partition, isolation group or system-wide
- *All* - privilege rank on the ExportRuleSet, or on ExportRuleSets in its partition, isolation group or system-wide

You always need *View* privileges on the partition of the ExportRuleSet.

See Also

- Privileges Required to use Exports

Privileges Required to use Export Rule Sets

Use

To use export jobs, you need either:

- `scheduler-administrator` role
- system, partition or isolation group-level permissions
- object-specific privileges

The privileges for the following objects are derived from ExportRuleSets:

- ExportRuleSets

Creating Export Jobs

You need the following privilege ranks to be able to create exports:

- *All* - full control over all Exports on the level the privilege was granted (system or isolation-group)
- *Submit* - on the *System_Export_RuleSet* job definition
- *View* - on the *GLOBAL*partition as well as on the *System* process server
- *JobAdministrator* - on the *System* queue.
- *View* - on the Cronacle Archive (CAR) format

See Also

- Privileges Required to use ExportRuleSets

Privileges Required to use Formats

Use

To use formats, you need either:

- `scheduler-administrator` role
- system, partition or isolation group-level permissions
- object-specific privileges

Creating Formats

You need one of the following privilege ranks to be able to create formats:

- *Create* - allows you to create formats on the level the privilege was granted (system, partition, isolation-group), you have no further privileges through this rank, you automatically get privileges on formats you create.
- *Edit* - allows you to create, view, and edit all formats on the level the privilege was granted (system, partition, isolation-group)
- *Delete* - allows you to create, view, and delete all formats on the level the privilege was granted (system, partition, isolation-group)
- *All* - full control over all formats on the level the privilege was granted (system, partition, isolation-group)

You always need *View* privileges on the partition in which you want to create the format.

Editing Formats

To successfully edit a format, you have to have one the following privileges:

- *Edit* - privilege rank on the format, or on formats in its partition, isolation group or system-wide
- *Delete* - privilege rank on the format, or on formats in its partition, isolation group or system-wide

- *All* - privilege rank on the format, or on formats in its partition, isolation group or system-wide

You always need *View* privileges on the partition of the format.

Formats can reference the following objects, you need at least *View* privileges on these objects and their partitions when you want to edit a format that references them:

- *Application*

See Also

- Privileges Required to use Applications

Privileges Required to use Email Alert Gateways

Use

To use email alert gateways, you need either:

- `scheduler-administrator` role
- system, partition or isolation group-level permissions
- object-specific privileges

Creating Email Alert Gateways

You need the following privilege ranks to be able to create email alert gateways:

- *Create* - allows you to create email alert gateways on the level the privilege was granted (system, partition, isolation-group), you have no further privileges through this rank, you automatically get privileges on email alert gateways you create.
- *Edit* - allows you to create, view and edit all email alert gateways on the level the privilege was granted (system, partition, isolation-group)
- *All* - full control over all email alert gateways on the level the privilege was granted (system, partition, isolation-group)

You always need *View* privileges on the partition in which you want to create the email alert gateway.

Editing Email Alert Gateways

- *Edit* - allows you to create, view and edit all email alert gateways on the level the privilege was granted (system, partition, isolation-group)
- *All* - full control over all email alert gateways on the level the privilege was granted (system, partition, isolation-group)

You always need *View* privileges on the partition of the email alert gateway.

Email alert gateways can reference the following objects, you need at least *View* privileges on these objects and their partitions when you want to edit an email alert gateway that references them:

- *Application*

See Also

- Privileges Required to use Applications

Privileges Required to use ImportRuleDefinitions

Use

To use ImportRuleDefinitions you need either:

- `scheduler-administrator` role
- system, partition or isolation group-level permissions
- object-specific privileges

Creating ImportRuleDefinitions

You need either of the following privilege ranks to be able to create ImportRuleDefinitions:

- *Create* - allows you to create ImportRuleDefinitions on the level the privilege was granted (system, partition, isolation-group), you have no further privileges through this rank, you automatically get privileges on ImportRuleDefinitions you create.
- *Edit* - allows you to create, view and edit all ImportRuleDefinitions on the level the privilege was granted (system, partition, isolation-group, object)
- *Delete* - allows you to create, view, and delete all ImportRuleDefinitions on the level the privilege was granted (system, partition, isolation-group, object)
- *All* - full control over all ImportRuleDefinitions on the level the privilege was granted (system, partition, isolation-group, object)

You always need *View* privileges on the partition in which you want to create the ImportRuleDefinition.

Editing ImportRuleDefinitions

To successfully edit an ImportRuleDefinition, you have to have at least one of the following privileges:

- *Edit* - privilege rank on the ImportRuleDefinition, or on ImportRuleDefinitions in its partition, isolation group or system-wide
- *All* - privilege rank on the ImportRuleDefinition, or on ImportRuleDefinitions in its partition, isolation group or system-wide

You always need *View* privileges on the partition of the ImportRuleDefinition.

See Also

- Privileges Required to use Imports

Privileges Required to use ImportRuleSets

Use

To use ImportRuleSets you need either:

- `scheduler-administrator` role
- system, partition or isolation group-level permissions
- object-specific privileges

Creating ImportRuleSets

You need either of the following privilege ranks to be able to create ImportRuleSets:

- *Create* - allows you to create ImportRuleSets on the level the privilege was granted (system, partition, isolation-group), you have no further privileges through this rank, you automatically get privileges on ExportRuleSets you create.
- *Edit* - allows you to create, view and edit all ImportRuleSets on the level the privilege was granted (system, partition, isolation-group, object)
- *Delete* - allows you to create, view, and delete all ImportRuleSets on the level the privilege was granted (system, partition, isolation-group, object)
- *All* - full control over all ImportRuleSets on the level the privilege was granted (system, partition, isolation-group, object)

You always need *View* privileges on the partition in which you want to create the ImportRuleSet.

Editing ImportRuleSets

To successfully edit an ImportRuleSet, you have to have at least one of the following privileges:

- *Edit* - privilege rank on the ImportRuleSet, or on ExportRuleSets in its partition, isolation group or system-wide
- *All* - privilege rank on the ImportRuleSet, or on ExportRuleSets in its partition, isolation group or system-wide

You always need *View* privileges on the partition of the ImportRuleSet.

See Also

- Privileges Required to use Imports

Privileges Required to use ImportSources

Use

To use ImportSources you need either:

- `scheduler-administrator` role
- system, partition or isolation group-level permissions
- object-specific privileges

Creating ImportSources

You need either of the following privilege ranks to be able to create ImportSources:

- *Create* - allows you to create ImportSources on the level the privilege was granted (system, partition, isolation-group), you have no further privileges through this rank, you automatically get privileges on ImportSources you create.
- *Edit* - allows you to create, view and edit all ImportSources on the level the privilege was granted (system, partition, isolation-group, object)

- *Delete* - allows you to create, view, and delete all ImportSources on the level the privilege was granted (system, partition, isolation-group, object)
- *All* - full control over all ImportSources on the level the privilege was granted (system, partition, isolation-group, object)

Editing ImportSources

To successfully edit an ImportSource, you have to have at least one of the following privileges:

- *Edit* - privilege rank on the ImportSource, or on ImportSources in its partition, isolation group or system-wide
- *All* - privilege rank on the ImportSource, or on ImportSources in its partition, isolation group or system-wide

See Also

- Privileges Required to use Imports

Privileges Required to use Import Rule Sets

Use

To use imports, you need either:

- `scheduler-administrator` role
- system, partition or isolation group-level permissions
- object-specific privileges

Privileges for the following objects are derived from the import privileges:

- Import
- ImportRuleDefinition
- ImportSource

Creating Imports

You need one of the following privilege ranks to be able to create imports:

- *All* - full control over all imports on the level the privilege was granted (system, partition, isolation-group)
- *Submit* - on the *System_Import_Archive* job definition
- *JobAdministrator*- on the *System* queue
- *View* - on the *GLOBAL*partition as well as on the *System* process server
- *View* - on the Cronacle Archive (CAR) format

Note

For each object you wish to import, you must have sufficient privileges to edit the object! This requires you to have at least *Edit* privileges on the object and *View* privileges on all other objects the object is referring to.

See Also

- Privileges Required to use Applications

Privileges Required to use Isolation Groups

Use

To create, edit, and delete isolation groups, you need the following role:

- `scheduler-isolation-administrator`

A user with this privilege is allowed no further privileges.

Privileges Required to use Job Chains

Use

To use job chains, you need either:

- `scheduler-administrator` role
- `scheduler-job-administrator` role
- system, partition or isolation group-level permissions
- object-specific privileges

See Privileges Required to use Job Definitions for more information.

Privileges Required to use Job Definition Alert Sources

Use

To use job definition alert sources, you need either:

- `scheduler-administrator` role
- system, partition or isolation group-level permissions
- object-specific privileges

Creating Job Definition Alert Sources

You need one of the following privilege ranks to be able to create job definition alert sources:

- *Create* - allows you to create job definition alert sources on the level the privilege was granted (system, partition, isolation-group), you have no further privileges through this rank, you automatically get privileges on job definition alert sources you create.
- *Edit* - allows you to create, view, and edit all job definition alert sources on the level the privilege was granted (system, partition, isolation-group)
- *Delete* - allows you to create, view, and delete all job definition alert sources on the level the privilege was granted (system, partition, isolation-group)
- *All* - full control over all job definition alert sources on the level the privilege was granted (system, partition, isolation-group)

You always need *View* privileges on the partition in which you want to create the job definition alert source.

Creating Alert Source Actions

You need the either of the following privilege ranks to be able to create ad hoc alert sources:

- *Create* - allows you to create an alert source action on the level the privilege was granted (system, partition, isolation-group), you have no further privileges through this rank, you automatically get privileges on ad hoc alert sources you create.
- *Edit* - allows you to create, view, and edit all alert source actions on the level the privilege was granted (system, partition, isolation-group)
- *Delete* - allows you to create, view, and delete all alert source actions on the level the privilege was granted (system, partition, isolation-group)
- *All* - full control over all alert source actions on the level the privilege was granted (system, partition, isolation-group)

Editing Job Definition Alert Sources

To successfully edit a job definition alert source, you have to have one the following privileges:

- *Edit* - privilege rank on the job definition alert source, or on job definition alert sources in its partition, isolation group or system-wide
- *Delete* - privilege rank on the job definition alert source, or on job definition alert sources in its partition, isolation group or system-wide
- *All* - privilege rank on the job definition alert source, or on job definition alert sources in its partition, isolation group or system-wide

You always need *View* privileges on the partition of the job definition alert source.

Job definition alert sources can reference the following objects, you need at least *View* privileges on these objects and their partitions when you want to edit a job definition that references them:

- *Application*
- *Escalation*
- *Time Window*

Editing Alert Source Actions

To successfully edit an alert source action, you have to have either of the following privileges:

- *Edit* - privilege rank on the alert source action, or on alert source actions in its partition, isolation group or system-wide
- *Delete* - privilege rank on the alert source action, or on alert source actions in its partition, isolation group or system-wide
- *All* - privilege rank on the alert source action, or on alert source actions in its partition, isolation group or system-wide

See Also

- Privileges Required to use Applications
- Privileges Required to use Applications

Privileges Required to use Job Definition Types

Use

To use Job Definition Types you need either:

- `scheduler-administrator` role
- system, partition or isolation group-level permissions
- object-specific privileges

Creating Job Definition Types

You need either of the following privilege ranks to be able to create job definition types:

- *Create* - allows you to create job definition types on the level the privilege was granted (system, partition, isolation-group), you have no further privileges through this rank, you automatically get privileges on job definition types you create.
- *Edit* - allows you to create, view and edit all job definition types on the level the privilege was granted (system, partition, isolation-group, object)
- *Delete* - allows you to create, view, and delete all job definition types on the level the privilege was granted (system, partition, isolation-group, object)
- *All* - full control over all job definition types on the level the privilege was granted (system, partition, isolation-group, object)

You always need *View* privileges on the partition in which you want to create the job definition type.

Editing Job Definition Types

To successfully edit an job definition type, you have to have at least one of the following privileges:

- *Edit* - privilege rank on the job definition type, or on job definition types in its partition, isolation group or system-wide
- *All* - privilege rank on the job definition type, or on job definition types in its partition, isolation group or system-wide

You always need *View* privileges on the partition of the job definition type.

Privileges Required to use Job Definitions

Use

To use job definitions, you need either:

- `scheduler-administrator` role
- `scheduler-job-administrator` role

- object, system, partition or isolation group-level permissions

The sections below will reference the object-level permissions you need, however, equivalent system, partition or isolation group-level permissions are also valid.

Creating Job Definitions

You need one of the following privilege ranks to be able to create job definitions:

- *Create* - allows you to create job definitions on the level the privilege was granted (system, partition, isolation-group), you have no further privileges through this rank, you automatically get privileges on job definitions you create.
- *Edit* - allows you to create, view, and edit all job definitions on the level the privilege was granted (system, partition, isolation-group)
- *Delete* - allows you to create, view, and delete all job definitions on the level the privilege was granted (system, partition, isolation-group)
- *All* - full control over all job definitions on the level the privilege was granted (system, partition, isolation-group)

You always need *View* privileges on the partition in which you want to create the job definition.

Submitting Job Definitions

To successfully submit a job definition, you have to have one of the following privileges:

- *Submit* - privilege rank on the job definition, or on job definitions in its partition, isolation group or system-wide
- *All* - privilege rank on the job definition, or on job definitions in its partition, isolation group or system-wide

As well as the following privileges

- *Submit Onto* - privilege rank on the queue where the job is supposed to run , or on queue where the job is supposed to run in its partition, isolation group or system-wide
- *View* - privilege rank on the process server that serves the queue and where the job is going to run, or on process server that serves the queue and where the job is going to runs in its partition, isolation group or system-wide

Optionally, if you want to use a library, submit frame and/or time window, you will need *View* privileges on them, as well. If the job definition has a default submit frame, a default time window, or uses a library (in parameters or in the code) you will require *View* privileges on the objects or you will not be able to submit the job definition or restart the job.

You always need *View* privileges on the partition of the job definition.

Job definitions can reference the following objects, you need at least *View* privileges on these objects when you want to submit a job definition that references them:

- *Credentials*
- *Events*
- *Locks*

Editing Job Definitions

To successfully edit a job definition, you have to have one the following privileges:

- *Edit* - privilege rank on the job definition, or on job definitions in its partition, isolation group or system-wide
- *Delete* - privilege rank on the job definition, or on job definitions in its partition, isolation group or system-wide
- *All* - privilege rank on the job definition, or on job definitions in its partition, isolation group or system-wide

Job definitions can reference the following objects, you need at least *View* privileges on these objects and their partitions when you want to edit a job definition that references them:

- *Library*
- *Credential*
- *Application*
- *Formats*
- *Locks*
- *Events*
- *Queues*
- *Submit Frame*
- *Time Window*
- *Resource*
- *QueryFilter* - used in job parameter constraints

See Also

- Privileges Required to use Applications
- Privileges Required to use Event Definitions
- Privileges Required to use Formats
- Privileges Required to use Credentials
- Privileges Required to use Libraries
- Privileges Required to use Locks
- Privileges Required to use Process Servers
- Privileges Required to use QueryFilters
- Privileges Required to use Queues
- Privileges Required to use Reports
- Privileges Required to use Resources

- Privileges Required to use Submit Frames
- Privileges Required to use Time Windows

Privileges Required to use Jobs

Use

To use jobs, you need either:

- `scheduler-administrator` role
- `scheduler-job-administrator` role
- object, system, partition or isolation group-level permissions

Viewing Jobs

To successfully view jobs, you have to have the following privileges:

- *View* - on the job definition, queue, process server, and their partition(s)
- *View* - privilege rank on Job in its partition, isolation group or system-wide

Note

The partition of the job is inherited from the queue where the job runs, not from the job definition!

Submitting and Restarting Jobs

To successfully submit a job, you have to have one of the following privileges:

- *Submit* - privilege rank on the job definition of the job, or on JobDefinition in its partition, isolation group or system-wide
- *All* - privilege rank on the job definition of the job, or on JobDefinition in its partition, isolation group or system-wide

You also require *View* privileges on the partition of the job definition, of the queue, and of the process server where the job has run and is going to run.

As well as the following privileges

- *JobAdministrator* - privilege rank on the queue where the job is supposed to run, or on either Queue in its partition, isolation group or system-wide
- *View* - privilege rank on the process server that serves the queue and where the job is going to run as well as its partition, or on ProcessServer in its partition, isolation group or system-wide
- *View* - privilege rank on the partitions of the process server, queue, and job definition

Optionally, if you want to use a library, submit frame and/or time window, you will need *View* privileges on them and their partitions, as well. If the job has a default submit frame, a default time window, or uses a library (in parameters or in the code) you will require *View* privileges on the objects or you will not be able to submit the job or restart the job.

Child-objects of a job such as a JobFile, for example, inherit the security from the job.

Jobs can reference the following objects, you need at least *View* privileges on these objects and their partition(s) when you want to submit a job that references them:

- *Credentials*
- *Events*
- *Library* - if the job uses code from the library.
- *Locks*

Deleting Jobs

All required privileges to see the job as well as the following:

- *JobAdministrator* - privilege rank on Job in its partition, isolation group or system-wide

Example

You have partitions named *P1*, *P2*, and *P3*; you want several users to be able to submit and monitor each other's jobs. The job definitions are all located in the `P1` partition, the process server *P3_ProcessServer* and queue *P2_Queue* are located in the *P3* and *P2* partitions, respectively, *P3_ProcessServer* serves *P2_Queue*. You do not want the users to see jobs in any other partitions.

Note

This example is an extreme case for illustration purposes; you would normally have the process server and the queue in the same partition.

You first create a group/role `p_operators` for the users in your external authentication system and grant it to users. If you are running NetWeaver you also assign the action *AccessScheduler* on the role.

1. Log on with a user who has the `p_operators` role so that it gets synchronized with SAP CPS.
2. Log on with a user that has *scheduler-administrator* privileges.
3. Navigate to *Security* > *Isolation Groups* or *Security* > *Partitions* (depending on your license, you will have one of the two).
4. Choose *Edit Security* from the context-menu of the *P1* partition.
5. Assign *p_operators* the *Granted Rank View*.
6. Choose *Save& Close*.
7. Choose *Edit Security* from the context-menu of the *P2* partition.
8. Assign *p_operators* the *Granted Rank View*.
9. Choose *Save& Close*.
10. Choose *Edit Security* from the context-menu of the *P3* partition.
11. Assign *p_operators* the *Granted Rank View*.
12. Choose *Save& Close*.
13. Navigate to *Security* > *Roles* and choose *Edit* from the context menu of the role *p_operators*.

14. On the *Privileges* tab, assign the privileges as the table indicates below.

15. Choose *Save & Close*.

Object Definition	Level	Level Object	Exportable	Granted Rank	Grantable Rank
Job	Partition	P2		View	None
Queue	Partition	P2		JobAdministrator	None
ProcessServer	Partition	P3		View	None
JobDefinition	Partition	P1		Submit	None

Note

Jobs inherit their partition from the queue, so the privilege on job must be in the same partition as the partition of the queue.

Note

Operators will not be allowed to delete jobs, if you want your operators to be able to delete jobs, assign them the *JobAdministrator* privilege in partition *P2* on *Job*. The table above reads *View* only for *Job* in partition *P2*, however, jobs can be restarted and resubmitted thanks to the *JobAdministrator* privilege on the queue (which is required to submit jobs).

Note

You can also save the three object privilege grants on the partitions above by adding a system-wide *View* privilege rank for partitions. However, users will be able to see all partitions.

See Also

- Privileges Required to use Event Definitions
- Privileges Required to use Credentials
- Privileges Required to use Libraries
- Privileges Required to use Locks
- Privileges Required to use Process Servers
- Privileges Required to use Queues
- Privileges Required to use Resources

Privileges Required to use Libraries

Use

To use libraries, you need either:

- `scheduler-administrator` role
- system, partition or isolation group-level permissions
- object-specific privileges

Creating Libraries

You need one of the following privilege ranks to be able to create libraries:

- *Create* - allows you to create libraries on the level the privilege was granted (system, partition, isolation-group), you have no further privileges through this rank, you automatically get privileges on libraries you create.
- *Edit* - allows you to create, view, and edit all libraries on the level the privilege was granted (system, partition, isolation-group)
- *Delete* - allows you to create, view, and delete all libraries on the level the privilege was granted (system, partition, isolation-group)
- *All* - full control over all libraries on the level the privilege was granted (system, partition, isolation-group)

You always need *View* privileges on the partition in which you want to create the library.

Editing Libraries

To successfully edit a library, you have to have one the following privileges:

- *Edit* - privilege rank on the library, or on libraries in its partition, isolation group or system-wide
- *Delete* - privilege rank on the library, or on libraries in its partition, isolation group or system-wide
- *All* - privilege rank on the library, or on libraries in its partition, isolation group or system-wide

You always need *View* privileges on the partition of the library.

Libraries can reference the following objects, you need at least *View* privileges on these objects and their partitions when you want to edit a library that references them:

- *Application*

See Also

- Privileges Required to use Applications

Privileges Required to use Locks

Use

To use locks, you need either:

- `scheduler-administrator` role
- system, partition or isolation group-level permissions
- object-specific privileges

Creating Locks

You need one of the following privilege ranks to be able to create locks:

- *Create* - allows you to create locks on the level the privilege was granted (system, partition, isolation-group), you have no further privileges through this rank, you automatically get privileges on locks you create.
- *Edit* - allows you to create, view, and edit all locks on the level the privilege was granted (system, partition, isolation-group)

- *Delete* - allows you to create, view, and delete all locks on the level the privilege was granted (system, partition, isolation-group)
- *All* - full control over all locks on the level the privilege was granted (system, partition, isolation-group)

You always need *View* privileges on the partition in which you want to create the lock.

Editing Locks

To successfully edit a lock, you have to have one the following privileges:

- *Edit* - privilege rank on the lock, or on locks in its partition, isolation group or system-wide
- *Delete* - privilege rank on the lock, or on locks in its partition, isolation group or system-wide
- *All* - privilege rank on the lock, or on locks in its partition, isolation group or system-wide

You always need *View* privileges on the partition of the lock.

Locks can reference the following objects, you need at least *View* privileges on these objects and their partitions when you want to edit a lock that references them:

- *Application*

See Also

- Privileges Required to use Applications

Privileges Required to use Monitor Nodes

Use

To use Monitor Nodes you need either:

- `scheduler-administrator` role
- system, partition or isolation group-level permissions
- object-specific privileges

Creating Monitor Nodes

You need either of the following privilege ranks to be able to create monitor nodes:

- *Create* - allows you to create monitor nodes on the level the privilege was granted (system, partition, isolation-group), you have no further privileges through this rank, you automatically get privileges on monitor nodes you create.
- *Edit* - allows you to create, view and edit all monitor nodes on the level the privilege was granted (system, partition, isolation-group, object)
- *Delete* - allows you to create, view, and delete all monitor nodes on the level the privilege was granted (system, partition, isolation-group, object)
- *All* - full control over all monitor nodes on the level the privilege was granted (system, partition, isolation-group, object)

Editing Monitor Nodes

To successfully edit an monitor node, you have to have at least one of the following privileges:

- *Edit* - privilege rank on the monitor node, or on monitor nodes in its partition, isolation group or system-wide
- *Operator*- allows you to view and confirm all monitor nodes on the level the privilege was granted (system, partition, isolation-group, object)
- *All* - privilege rank on the monitor node, or on monitor nodes in its partition, isolation group or system-wide

Privileges Required to use Operator Messages

Use

To use Operator Messages you need either:

- `scheduler-administrator` role
- `scheduler-job-administrator` role
- system, partition or isolation group-level permissions
- object-specific privileges

Creating Operator Messages

You need either of the following privilege ranks to be able to create operator messages:

- *Operator* - allows you to view and reply to all operator messages on the level the privilege was granted (system, partition, isolation-group, object)
- *Delete* - allows you to view and delete all operator messages on the level the privilege was granted (system, partition, isolation-group, object)
- *All* - full control over all operator messages on the level the privilege was granted (system, partition, isolation-group, object)

Editing Operator Messages

To successfully reply to an operator message, you have to have at least one of the following privileges:

- *Operator* - privilege rank on the operator message, or on operator messages in its partition, isolation group or system-wide
- *All* - privilege rank on the operator message, or on operator messages in its partition, isolation group or system-wide

See Also

- Privileges Required to use Exports

Privileges Required to use Partitions

Use

To use partitions, you need either:

- `scheduler-administrator` role

- system or isolation group-level permissions
- object-specific privileges

Creating Partitions

You need one of the following privilege ranks to be able to create partitions:

- *Create* - allows you to create partitions on the level the privilege was granted (system, isolation-group), you have no further privileges through this rank, you automatically get privileges on partitions you create.
- *Edit* - allows you to create, view, and edit all partitions on the level the privilege was granted (system, isolation-group)
- *Delete* - allows you to create, view, and delete all partitions on the level the privilege was granted (system, isolation-group)
- *All* - full control over all partitions on the level the privilege was granted (system, isolation-group)

Editing Partitions

To successfully edit a partition, you have to have one the following privileges:

- *Edit* - privilege rank on the partition, or on partitions in its isolation group or system-wide
- *Delete* - privilege rank on the partition, or on partitions in its isolation group or system-wide
- *All* - privilege rank on the partition, or on partitions in its isolation group or system-wide

Privileges Required to use Period Functions

Use

To use period functions, you need either:

- `scheduler-administrator` role
- system, partition or isolation group-level permissions
- object-specific privileges

Creating Period Functions

You need one of the following privilege ranks to be able to create period functions:

- *Create* - allows you to create period functions on the level the privilege was granted (system, partition, isolation-group), you have no further privileges through this rank, you automatically get privileges on period functions you create.
- *Edit* - allows you to create, view, and edit all period functions on the level the privilege was granted (system, partition, isolation-group)
- *Delete* - allows you to create, view, and delete all period functions on the level the privilege was granted (system, partition, isolation-group)
- *All* - full control over all period functions on the level the privilege was granted (system, partition, isolation-group)

You always need *View* privileges on the partition in which you want to create the period function.

Editing Period Functions

To successfully edit a period function, you have to have one the following privileges:

- *Edit* - privilege rank on the period function, or on period functions in its partition, isolation group or system-wide
- *Delete* - privilege rank on the period function, or on period functions in its partition, isolation group or system-wide
- *All* - privilege rank on the period function, or on period functions in its partition, isolation group or system-wide

You always need *View* privileges on the partition of the period function.

Period functions can reference the following objects, you need at least *View* privileges on these objects and their partitions when you want to edit a period function that references them:

- *Library*
- *Application*

See Also

- Privileges Required to use Applications

15,0-1 58%

Privileges Required to use Process Server Alert Sources

Use

To use process server alert sources, you need either:

- `scheduler-administrator` role
- system, partition or isolation group-level permissions
- object-specific privileges

Creating Process Server Alert Sources

You need one of the following privilege ranks to be able to create process server alert sources:

- *Create* - allows you to create process server alert sources on the level the privilege was granted (system, partition, isolation-group), you have no further privileges through this rank, you automatically get privileges on process server alert sources you create.
- *Edit* - allows you to create, view, and edit all process server alert sources on the level the privilege was granted (system, partition, isolation-group)
- *Delete* - allows you to create, view, and delete all process server alert sources on the level the privilege was granted (system, partition, isolation-group)
- *All* - full control over all process server alert sources on the level the privilege was granted (system, partition, isolation-group)

You always need *View* privileges on the partition in which you want to create the process server alert source.

Editing Process Server Alert Sources

To successfully edit a process server alert source, you have to have one the following privileges:

- *Edit* - privilege rank on the process server alert source, or on process server alert sources in its partition, isolation group or system-wide
- *Delete* - privilege rank on the process server alert source, or on process server alert sources in its partition, isolation group or system-wide
- *All* - privilege rank on the process server alert source, or on process server alert sources in its partition, isolation group or system-wide

You always need *View* privileges on the partition of the process server alert source.

Process server alert sources can reference the following objects, you need at least *View* privileges on these objects and their partitions when you want to edit a process server alert source that references them:

- *Application*
- *Process Server*

Editing Alert Source Actions

To successfully edit a ad hoc alert source, you have to have either of the following privileges:

- *Edit* - privilege rank on the alert source action, or on alert source actions in its partition, isolation group or system-wide
- *Delete* - privilege rank on the alert source action, or alert source actions in its partition, isolation group or system-wide
- *All* - privilege rank on the alert source action, or on alert source actions in its partition, isolation group or system-wide

See Also

- Privileges Required to use Applications
- Privileges Required to use Process Servers

Privileges Required to use Process Servers

Use

To use process servers, you need either:

- `scheduler-administrator` role
- system, partition or isolation group-level permissions
- object-specific privileges

Creating Process Severs

You need one of the following privilege ranks to be able to create process severs:

- *Create* - allows you to create process severs on the level the privilege was granted (system, partition, isolation-group), you have no further privileges through this rank, you automatically get privileges on process severs you create.
- *Edit* - allows you to create, view, and edit all process severs on the level the privilege was granted (system, partition, isolation-group)
- *Delete* - allows you to create, view, and delete all process severs on the level the privilege was granted (system, partition, isolation-group)
- *All* - full control over all process severs on the level the privilege was granted (system, partition, isolation-group)

You always need *View* privileges on the partition in which you want to create the process server.

Editing Process Severs

To successfully edit a process sever, you have to have one the following privileges:

- *Edit* - privilege rank on the process sever, or on process severs in its partition, isolation group or system-wide
- *Delete* - privilege rank on the process sever, or on process severs in its partition, isolation group or system-wide
- *All* - privilege rank on the process sever, or on process severs in its partition, isolation group or system-wide

You always need *View* privileges on the partition of the process server.

Process severs can reference the following objects, you need at least *View* privileges on these objects and their partitions when you want to edit a process sever that references them:

- *Application*
- *Queues*

Deleting Process Servers

To successfully delete a process sever, you have to have one the following privileges:

- *Delete* - privilege rank on the process sever, or on process severs in its partition, isolation group or system-wide
- *All* - privilege rank on the process sever, or on process severs in its partition, isolation group or system-wide

You must also make sure that there are no jobs left that ran on the process server and no file events on the process server left.

You always need *View* privileges on the partition of the process server.

See Also

- Privileges Required to use Applications
- Privileges Required to use Queues

Privileges Required to use QueryFilters

Use

To use object filters, you need either:

- `scheduler-administrator` role
- object, system, partition or isolation group-level permissions

The sections below will reference the object-level permissions you need, however, equivalent system, partition or isolation group-level permissions are also valid.

Creating Query Filters

You need one of the following privilege ranks to be able to create object filters:

- *Create* - on QueryFilter allows you to create object filters
- *Edit* - on QueryFilter allows you to create, view, and edit all object filters
- *Delete* - on QueryFilter allows you to create, view, and delete all object filters
- *All* - on QueryFilter grants full control over all object filters

Editing Query Filters

To successfully edit a object filter, you have to have one the following privileges:

- *Edit* - system-wide privilege rank on QueryFilter
- *Delete* - system-wide privilege rank on QueryFilter
- *All* - system-wide privilege rank on QueryFilter

Privileges Required to use Queue Alert Source

Use

Queue alert sources are not available.

Privileges Required to use Queues

Use

To use queues, you need either:

- `scheduler-administrator` role
- system, partition or isolation group-level permissions
- object-specific privileges

Creating Queues

You need one of the following privilege ranks to be able to create queues:

- *Create* - allows you to create queues on the level the privilege was granted (system, partition, isolation-group), you have no further privileges through this rank, you automatically get privileges on queues you create.

- *Edit* - allows you to create, view, and edit all queues on the level the privilege was granted (system, partition, isolation-group)
- *Delete* - allows you to create, view, and delete all queues on the level the privilege was granted (system, partition, isolation-group)
- *All* - full control over all queues on the level the privilege was granted (system, partition, isolation-group)

You always need *View* privileges on the partition of the queue.

Editing Queues

To successfully edit a queue, you have to have one the following privileges:

- *Edit* - privilege rank on the queue, or on queues in its partition, isolation group or system-wide
- *Delete* - privilege rank on the queue, or on queues in its partition, isolation group or system-wide
- *All* - privilege rank on the queue, or on queues in its partition, isolation group or system-wide

You always need *View* privileges on the partition of the queue.

Queues can reference the following objects, you need at least *View* privileges on these objects and their partitions when you want to edit a queue that references them:

- *Application*
- *Submit Frame*
- *Time Window*
- *Process Servers* - A queue might appear held to you while it is open to others; this is usually due to the fact that you cannot see any of the running process servers that are serving the queue.

See Also

- Privileges Required to use Applications
- Privileges Required to use Process Servers
- Privileges Required to use Submit Frames
- Privileges Required to use Time Windows

Privileges Required to use Registry Entries

Use

To use registry entries, you need either:

- `scheduler-administrator` role
- system, partition or isolation group-level permissions
- object-specific privileges

Creating Registry Entries

You need one of the following privilege ranks to be able to create registry entries:

- *Create* - allows you to create registry entries on the level the privilege was granted (system, partition, isolation-group), you have no further privileges through this rank, you automatically get privileges on registry entries you create.
- *Edit* - allows you to create, view, and edit all registry entries on the level the privilege was granted (system, partition, isolation-group)
- *Delete* - allows you to create, view, and delete all registry entries on the level the privilege was granted (system, partition, isolation-group)
- *All* - full control over all registry entries on the level the privilege was granted (system, partition, isolation-group)

Editing Registry Entries

To successfully edit a registry entry, you have to have one the following privileges:

- *Edit* - privilege rank on the registry entry, or on registry entries in its partition, isolation group or system-wide
- *Delete* - privilege rank on the registry entry, or on registry entries in its partition, isolation group or system-wide
- *All* - privilege rank on the registry entry, or on registry entries in its partition, isolation group or system-wide

Registry entries can reference the following objects, you need at least *View* privileges on these objects and their partitions when you want to edit a registry entry that references them:

- *Library*
- *Credential*
- *Application*
- *Formats*
- *Locks*
- *Events*
- *Queues*
- *Submit Frame*
- *Time Window*
- *Resource*

See Also

- Privileges Required to use Applications
- Privileges Required to use Event Definitions
- Privileges Required to use Formats
- Privileges Required to use Credentials
- Privileges Required to use Libraries

- Privileges Required to use Locks

Privileges Required to use Reports

Use

To use reports, you need either:

- `scheduler-administrator` role
- system, partition or isolation group-level permissions
- object-specific privileges

Creating Reports

You need one of the following privilege ranks to be able to create reports:

- *Create* - allows you to create reports on the level the privilege was granted (system, partition, isolation-group), you have no further privileges through this rank, you automatically get privileges on reports you create.
- *Edit* - allows you to create, view, and edit all reports on the level the privilege was granted (system, partition, isolation-group)
- *Delete* - allows you to create, view, and delete all reports on the level the privilege was granted (system, partition, isolation-group)
- *All* - full control over all reports on the level the privilege was granted (system, partition, isolation-group)

You always need *View* privileges on the partition of the report.

Editing Reports

To successfully edit a report, you have to have one the following privileges:

- *Edit* - privilege rank on the report, or on reports in its partition, isolation group or system-wide
- *Delete* - privilege rank on the report, or on reports in its partition, isolation group or system-wide
- *All* - privilege rank on the report, or on reports in its partition, isolation group or system-wide

Reports can reference the following objects, you need at least *View* privileges on these objects and their partitions when you want to edit a report that references them:

- *Application*

You always need *View* privileges on the partition of the report.

See Also

- Privileges Required to use Applications

Privileges Required to use Resources

Use

To use resources, you need either:

- `scheduler-administrator` role

- system, partition or isolation group-level permissions
- object-specific privileges

Creating Resources

You need one of the following privilege ranks to be able to create resources:

- *Create* - allows you to create resources on the level the privilege was granted (system, partition, isolation-group), you have no further privileges through this rank, you automatically get privileges on resources you create.
- *Edit* - allows you to create, view, and edit all resources on the level the privilege was granted (system, partition, isolation-group)
- *Delete* - allows you to create, view, and delete all resources on the level the privilege was granted (system, partition, isolation-group)
- *All* - full control over all resources on the level the privilege was granted (system, partition, isolation-group)

You always need *View* privileges on the partition of the resource.

Editing Resources

To successfully edit a resource, you have to have one the following privileges:

- *Edit* - privilege rank on the resource, or on resources in its partition, isolation group or system-wide
- *Delete* - privilege rank on the resource, or on resources in its partition, isolation group or system-wide
- *All* - privilege rank on the resource, or on resources in its partition, isolation group or system-wide

You always need *View* privileges on the partition of the resource.

Resources can reference the following objects, you need at least *View* privileges on these objects and their partitions when you want to edit a resource that references them:

- *Application*
- *Process Server*

See Also

- Privileges Required to use Applications
- Privileges Required to use Process Servers

Privileges Required to use Roles

Use

Roles are hosted in external authentication systems and cannot be edited from within SAP CPS.

Users can grant roles privileges they have with an '"Admin" rank to custom roles. Custom roles are roles that did not ship with SAP CPS.

Privileges Required to use SAP Systems

Use

To use SAP systems, you need either:

- `scheduler-administrator` role
- system, partition or isolation group-level permissions
- object-specific privileges

Creating SAP Systems

You need one of the following privilege ranks to be able to create SAP systems:

- *Create* - allows you to create SAP systems on the level the privilege was granted (system, partition, isolation-group), you have no further privileges through this rank, you automatically get privileges on SAP systems you create.
- *Edit* - allows you to create, view, and edit all SAP systems on the level the privilege was granted (system, partition, isolation-group)
- *Delete* - allows you to create, view, and delete all SAP systems on the level the privilege was granted (system, partition, isolation-group)
- *All* - full control over all SAP systems on the level the privilege was granted (system, partition, isolation-group)

You always need *View* privileges on the partition of the SAP system.

Editing SAP Systems

To successfully edit a SAP system, you have to have one the following privileges:

- *Edit* - privilege rank on the SAP system, or on SAP systems in its partition, isolation group or system-wide
- *Delete* - privilege rank on the SAP system, or on SAP systems in its partition, isolation group or system-wide
- *All* - privilege rank on the SAP system, or on SAP systems in its partition, isolation group or system-wide

You always need *View* privileges on the partition of the SAP system.

SAP systems can reference the following objects, you need at least *View* privileges on these objects and their partitions when you want to edit a SAP system that references them:

- *Application*
- *Events*

Using the SAP CPS RFC Server

A special rank allows you to use the RFC server for Closing Cockpit, Solution Manager integration and use of other SAP technologies.

- *Business Automation API* - allows you to interact with SAP CPS from within an ABAP stack of an SAP instance.

See Also

- Privileges Required to use Applications

- Privileges Required to use Event Definitions

Privileges Required to use Services

Use

Services are viewable to every SAP CPS user, you cannot change of delete them, so no privilege ranks are available for Service.

Privileges Required to use the Shell

Use

To use the shell, you need the following *Global Privilege* :

- *System_Shell* - allows you to access the command-line interface known as the *Shell*.

Privileges Required to use StatisticCagtegories

Use

StatisticCagtegories are used to calculate job statistics.

To use StatisticCagtegories, you need either:

- `scheduler-administrator` role
- system, partition or isolation group-level permissions
- object-specific privileges

Creating StatisticCagtegories

You need the either of the following privilege ranks to be able to create StatisticCagtegories:

- *Create* - allows you to create an ad StatisticCagtegory on the level the privilege was granted (system, partition, isolation-group), you have no further privileges through this rank, you automatically get privileges on StatisticCagtegories you create.
- *Edit* - allows you to create, view, and edit all StatisticCagtegories on the level the privilege was granted (system, partition, isolation-group)
- *Delete* - allows you to create, view, and delete all ad StatisticCagtegory on the level the privilege was granted (system, partition, isolation-group)
- *All* - full control over all StatisticCagtegories on the level the privilege was granted (system, partition, isolation-group)

Editing StatisticCagtegories

To successfully edit a ad StatisticCagtegory, you have to have either of the following privileges:

- *Edit* - privilege rank on the ad StatisticCagtegory, or on StatisticCagtegories in its partition, isolation group or system-wide
- *Delete* - privilege rank on the ad StatisticCagtegory, or on a StatisticCagtegory in its partition, isolation group or system-wide

- *All* - privilege rank on the ad StatisticCagtegory, or on StatisticCagtegories in its partition, isolation group or system-wide

See Also

StatisticCategory

Privileges Required to use SubjectIsolationGroups

Use

SubjectIsolationGroups are used to calculate job statistics.

To use SubjectIsolationGroups, you need either:

- `scheduler-administrator` role
- system, partition or isolation group-level permissions
- object-specific privileges

Creating SubjectIsolationGroups

You need the either of the following privilege ranks to be able to create SubjectIsolationGroups:

- *Create* - allows you to create an SubjectIsolationGroup on the level the privilege was granted (system, partition, isolation-group), you have no further privileges through this rank, you automatically get privileges on SubjectIsolationGroups you create.
- *Edit* - allows you to create, view, and edit all SubjectIsolationGroups on the level the privilege was granted (system, partition, isolation-group)
- *Delete* - allows you to create, view, and delete all SubjectIsolationGroups on the level the privilege was granted (system, partition, isolation-group)
- *All* - full control over all SubjectIsolationGroups on the level the privilege was granted (system, partition, isolation-group)

Editing SubjectIsolationGroups

To successfully edit a SubjectIsolationGroup, you have to have either of the following privileges:

- *Edit* - privilege rank on the SubjectIsolationGroup, or on SubjectIsolationGroups in its partition, isolation group or system-wide
- *Delete* - privilege rank on the SubjectIsolationGroup, or on SubjectIsolationGroups in its partition, isolation group or system-wide
- *All* - privilege rank on the SubjectIsolationGroup, or on SubjectIsolationGroups in its partition, isolation group or system-wide

See Also

StatisticCategory

Privileges Required to use SubjectRoleGrantExpanded

Use

The SubjectRoleGrantExpanded property is not used at this moment.

Privileges Required to use Subjects

Use

To use Subjects you need either:

- `scheduler-administrator` role
- system-level permissions

Creating Subjects

You need either of the following privilege ranks to be able to create subjects:

- *Manage* - allows you to enable all subjects, system-wide.
- *All* - allows you to enable all subjects, system-wide.

Privileges Required to use Submit Frames

Use

To use submit frames, you need either:

- `scheduler-administrator` role
- `scheduler-job-administrator`
- system, partition or isolation group-level permissions
- object-specific privileges

Creating Submit Frames

You need one of the following privilege ranks to be able to create submit frames:

- *Create* - allows you to create submit frames on the level the privilege was granted (system, partition, isolation-group), you have no further privileges through this rank, you automatically get privileges on submit frames you create.
- *Edit* - allows you to create, view, and edit all submit frames on the level the privilege was granted (system, partition, isolation-group)
- *Delete* - allows you to create, view, and delete all submit frames on the level the privilege was granted (system, partition, isolation-group)
- *All* - full control over all submit frames on the level the privilege was granted (system, partition, isolation-group)

You always need *View* privileges on the partition of the submit frame.

Editing Submit Frames

To successfully edit a submit frame, you have to have one the following privileges:

- *Edit* - privilege rank on the submit frame, or on submit frames in its partition, isolation group or system-wide
- *Delete* - privilege rank on the submit frame, or on submit frames in its partition, isolation group or system-wide
- *All* - privilege rank on the submit frame, or on submit frames in its partition, isolation group or system-wide

You always need *View* privileges on the partition of the submit frame.

Submit frames can reference the following objects, you need at least *View* privileges on these objects and their partitions when you want to edit a submit frame that references them:

- *Application*
- *Time Window*

See Also

- Privileges Required to use Applications
- Privileges Required to use Time Windows

Privileges Required to use Table Definitions

Use

To use table definitions, you need either:

- `scheduler-administrator` role
- system, partition or isolation group-level permissions
- object-specific privileges

Creating Table Definitions

You need one of the following privilege ranks to be able to create table definitions:

- *Create* - allows you to create table definitions on the level the privilege was granted (system, partition, isolation-group), you have no further privileges through this rank, you automatically get privileges on table definitions you create.
- *Edit* - allows you to create, view, and edit all table definitions on the level the privilege was granted (system, partition, isolation-group)
- *Delete* - allows you to create, view, and delete all table definitions on the level the privilege was granted (system, partition, isolation-group)
- *All* - full control over all table definitions on the level the privilege was granted (system, partition, isolation-group)

You always need *View* privileges on the partition of the table definition.

Editing Table Definitions

To successfully edit a table definition, you have to have one the following privileges:

- *Edit* - privilege rank on the table definition, or on table definitions in its partition, isolation group or system-wide
- *Delete* - privilege rank on the table definition, or on table definitions in its partition, isolation group or system-wide
- *All* - privilege rank on the table definition, or on table definitions in its partition, isolation group or system-wide

You always need *View* privileges on the partition of the table definition.

Table definitions can reference the following objects, you need at least *View* privileges on these objects and their partitions when you want to edit a table definition that references them:

- *Application*

See Also

- Privileges Required to use Applications

Privileges Required to use Tables

Use

To use tables, you need either:

- `scheduler-administrator` role
- system, partition or isolation group-level permissions
- object-specific privileges

Creating Tables

You need one of the following privilege ranks to be able to create tables:

- *Create* - allows you to create tables on the level the privilege was granted (system, partition, isolation-group), you have no further privileges through this rank, you automatically get privileges on tables you create.
- *Edit* - allows you to create, view, and edit all tables on the level the privilege was granted (system, partition, isolation-group)
- *Delete* - allows you to create, view, and delete all tables on the level the privilege was granted (system, partition, isolation-group)
- *All* - full control over all tables on the level the privilege was granted (system, partition, isolation-group)

You always need *View* privileges on the partition of the table.

Editing Tables

To successfully edit a table, you have to have one the following privileges:

- *Edit* - privilege rank on the table, or on tables in its partition, isolation group or system-wide

- *Delete* - privilege rank on the table, or on tables in its partition, isolation group or system-wide
- *All* - privilege rank on the table, or on tables in its partition, isolation group or system-wide

You always need *View* privileges on the partition of the table.

Tables can reference the following objects, you need at least *View* privileges on these objects and their partitions when you want to edit a table that references them:

- *Application*

See Also

- Privileges Required to use Applications

Privileges Required to use Time Windows

Use

To use time windows, you need either:

- `scheduler-administrator` role
- `scheduler-job-administrator`
- system, partition or isolation group-level permissions
- object-specific privileges

Creating Time Windows

You need one of the following privilege ranks to be able to create time windows:

- *Create* - allows you to create time windows on the level the privilege was granted (system, partition, isolation-group), you have no further privileges through this rank, you automatically get privileges on time windows you create.
- *Edit* - allows you to create, view, and edit all time windows on the level the privilege was granted (system, partition, isolation-group)
- *Delete* - allows you to create, view, and delete all time windows on the level the privilege was granted (system, partition, isolation-group)
- *All* - full control over all time windows on the level the privilege was granted (system, partition, isolation-group)

You always need *View* privileges on the partition of the time window.

Editing Time Windows

To successfully edit a time window, you have to have one the following privileges:

- *Edit* - privilege rank on the time window, or on time windows in its partition, isolation group or system-wide
- *Delete* - privilege rank on the time window, or on time windows in its partition, isolation group or system-wide
- *All* - privilege rank on the time window, or on time windows in its partition, isolation group or system-wide

You always need *View* privileges on the partition of the time window.

Time windows can reference the following objects, you need at least *View* privileges on these objects and their partitions when you want to edit a time window that references them:

- *Application*
- *Period Function*
- *Submit Frame*
- *Time Window*
- *Resource*

See Also

- Privileges Required to use Applications
- Privileges Required to use Submit Frames
- Privileges Required to use Period Functions

Privileges Required to use Time Zones

Use

Time zones are viewable by all SAP CPS users.

Privileges Required to use Triggers

Use

To use triggers, you need either:

- `scheduler-administrator` role
- system, partition or isolation group-level permissions
- object-specific privileges

Creating Triggers

You need one of the following privilege ranks to be able to create triggers:

- *Create* - allows you to create triggers on the level the privilege was granted (system, partition, isolation-group), you have no further privileges through this rank, you automatically get privileges on triggers you create.
- *Edit* - allows you to create, view, and edit all triggers on the level the privilege was granted (system, partition, isolation-group)
- *Delete* - allows you to create, view, and delete all triggers on the level the privilege was granted (system, partition, isolation-group)
- *All* - full control over all triggers on the level the privilege was granted (system, partition, isolation-group)

You always need *View* privileges on the partition of the trigger.

Editing Triggers

To successfully edit a trigger, you have to have one the following privileges:

- *Edit* - privilege rank on the trigger, or on triggers in its partition, isolation group or system-wide
- *Delete* - privilege rank on the trigger, or on triggers in its partition, isolation group or system-wide
- *All* - privilege rank on the trigger, or on triggers in its partition, isolation group or system-wide

You always need *View* privileges on the partition of the trigger,

Triggers can reference the following objects, you need at least *View* privileges on these objects and their partitions when you want to edit a trigger that references them:

- *Library*
- *Application*

See Also

- Privileges Required to use Applications
- Privileges Required to use Libraries

Privileges Required to use Users

Use

Users are hosted in external authentication systems and cannot be edited from within SAP CPS.

Users can grant privileges they have with an *Admin* rank to other users.

Auditing Object Changes

Use

Objects can be changed by users and this is a potential cause of havoc. To allow you to trace and easily revert changes, the auditing module was introduced. The auditing module allows you to set up rules which will either trace changes on objects only, referred to as *Diff only*, or trace changes and revert, referred to as *Full auditing*.

Note

The Active Auditing Module requires the *Module.Auditing* license key to be present in your license.

Note

Under certain circumstances, you will receive audits for objects you are not directly auditing. For example, if you audit submit frames only and add a submit frame element which is defined by a time window, you will receive audits for this time window as well as the submit frame. When you change the time window, you change the submit frame, indirectly.

Object auditing does not apply when the system makes a change.

Examples of where the system makes a change are:

- Automatically submitted job definitions, with wait events for example
- Changes to configuration as the result of running system jobs the *System_ConfigureMail* or *SAP_ImportCcmsMonitors* for example

When an audited object has been modified, it will get a new tab in the editor named *Audit trail*, which allows you to quickly undo any changes.

Note

The default auditing rule on audit objects cannot be modified. It is there to audit all changes to auditing objects, so nobody can escape auditing. You cannot change an audit object and not get caught.

Actions

Audit Rules support the following actions:

Action	**Description**
Export	Export the audit rule into a CAR file
Edit	Edit the audit rule
Edit Security	Edit the security of the audit rule
Delete	Delete the audit rule
Duplicate	Make a copy of the audit rule to create a similar one
Expand All	Expand all audit rules in the current filter
New	Create a new audit rule
Filter > New Filter	Create a new audit rule filter
Filter > Edit Filter	Edit current audit rule filter
Filter > Delete	Delete current audit rule filter

The actions are available for audit entries in the *Audit Trail*:

Action	**Description**
Restore to before change	Undo the selected and all subsequent changes
Restore to after change	Undo all subsequent changes to the object; in other words restore the object to the state it was after the selected change
Expand All	Expand all audit trail entries in the current filter
Filter > New Filter	Create a new audit trail filter
Filter > Edit Filter	Edit current audit trail filter
Filter > Delete	Delete current audit trail filter

Finding Audit Entries

You can search for audit entries using the *Search Audit Entries* box located under your username on the top right-hand side of the user interface. This is known as intelliSearch and allows you to specify complex queries in a simple way using prefixes. Prefixes are used to specify which property you are searching in and have short and long syntaxes. For example, if you want to display all audit entries created between 2:05 PM and 2:06 PM, you would use the search criteria as follows:

```
t:14:05-14:06
```

You can search more than one property, as follows:

```
t:14:05-14:06 u:jdoe
```

Note

No spaces should be entered before or after the colon (:).

See the Advanced Object Search for more information.

The following table illustrates the available prefixes for audit entries:

Prefix	Description
t, time	creation time
o, object objecttype	object type (case sensitive)
k, key businesskey	business key
a, action	action, specify the code C=Created, M=Modified, D=Deleted
u, user	user
l, level	audit level, specify the level 1=Diff only, 2=Full audit

There are different ways of searching by time:

- *hh:mm* - without a range, the default range is +/- 15 minutes.
- *hh:mm:ss* - without a range, the default range is +/- 1 minute.
- *hh:mm-hh:mm* or *hh:mm:ss-hh:mm:ss* - range is from start time to end time.

Reverting a change

You can freely revert a change and, even if this was a mistake, you can revert back to the change again. The following actions have been introduced:

- *Restore to before change*
- *Restore to after change*

Note

These two actions are also available via scripting via the methods `restoreBefore` and `restoreAfter` on the `AuditObject` object.

These actions are only visible if there is a record for before and after the change respectively:

- *Object Created* - only Restore to after
- *Object Modified* - Both before and after
- *Object Deleted* - only Restore to before

Security

Privilege	Description
AuditingRule.Create	Create auditing rules

AuditingRule.Delete	Delete auditing rules
AuditingRule.Edit	Edit auditing rules
AuditingRule.View	Access auditing rules

You can grant privileges on two levels, *Access* and *Admin;* a privilege granted on *Admin* level allows the grantee to grant the privilege to other users. These privileges can be granted *system-wide*, per *partition* or *isolation group*.

If you have the security module, which requires the *Module.Security* license key, you have an additional *Security* tab on the auditing rule. It allows you to specify which users can access, edit, and delete the auditing rule.

- Granting and Revoking System Privileges
- Granting or Revoking Object Privileges

Procedure

Create an audit rule

1. Navigate to *Auditing* > *Audit Rules*.
2. Choose *New* from the context menu.
3. Select an object type in the *Rule Object Type* and a *Level*.
4. Specify optional match criteria, refer to the *Values* section below.
5. Choose *Save & Close*.

Revert a change

1. Navigate to *Auditing* > *Audit Trail*.
2. Choose *Revert to before change* from the context menu of the audit entry you would like to revert.

Values

Field	**Description**	**Values**
Rule Object Type	The type of object you want to audit	
Level	The level of auditing you want, full audit allows you to revert changes	Diff Only Full Audit
Name Reg Ex	(optional) A regular expression pattern that is used to match object names	
Application	(optional) The application the object resides in	

Application Rule	Rule that allows you to match objects based on their application.	All Applications - Will match an object if it has an application or not Exact Application - Will match objects which have the application specified (for this option, the *Application* field is mandatory) No Application - Will match an object, if it has no application Any Application - Will match an object, if it has an application Sub Application - Will match an object, if its application is a child application of the application specified in the *Application* field (for this option, the *Application* field is mandatory)
Enabled	When this is checked, the rule is enabled.	

Example

Revert changes made to a job chain

1. Navigate to *Auditing > Audit Rules*.
2. Choose *New* from the context menu.
3. Select *Job Chain* as the *Rule Object Type* and *Full Audit* as the *Level*.
4. Fill `JCprdFin.*` into the *Name Reg Ex* field.
5. Fill `Finance` into the *Application* field.
6. Select *Exact Application* in the *Application Rule* drop-down.
7. Choose *Save & Close*.
8. Navigate to *Definitions > Job Chains*.
9. Choose *New*, in the new window choose the *Job Definition* tab and fill `JCprdFinQtrRep` into the *Name* field.
10. Choose *Save*.
11. Leave the editor window without closing it, return to the main window and navigate to *Auditing > Audit Trail*. Notice the new entry for for the creation of the job chain, choose *Refresh* from the context menu if you do not see the new entry.
12. Return to the job chain editor window, add a step and a job with the *System_Info* job definition.
13. Choose *Save & Close*.
14. In the main window, notice the new entries in *Audit Trail*, choose *Revert to before change* from the context menu of the *Modified* entry of the job chain.
15. Navigate to *Definitions > Job Chains* and inspect it, notice that the step and the job have vanished.

Generating Reports

Use

Reports allow you to gather meta-data about your repository in one of the following formats:

- Comma separated values (CSV) to import into your favorite spreadsheet application
- HTML to view in your browser

You can define multiple reports, and schedule *System_ReportRun* to generate the output. You can generate a report based on an object filter or a query on the *Report* tab, choose the columns of the query you want to end up in the report in the *Columns* tab, and throttle the amount of rows in the *Selections* tab based on the values of two columns or the value in a column. The default filter names are descriptive, for example, the *.alltop* filters will only return data about the top nodes of applications.

Note

Your custom object filters are available by their name, which might differ from the description which is shown in the object view.

Actions

Reports support the following actions:

Action	**Description**
Submit Report	Submit a job to generate the report
Export	Export the report into a CAR file
Edit	Edit the report
Edit Security	Edit the security of the report
Delete	Delete the report
Duplicate	Make a copy of the report to create a similar one
Expand All	Expand all reports in the current filter
New	Create a new report
Filter > New Filter	Create a new report filter
Filter > Edit Filter	Edit current report filter
Filter > Delete	Delete current report filter

Finding Reports

You can search for reports using filters and the *Search Reports* box on the Reports tab. This box is known as the *intelliSearch* box and located under your username on the top right-hand side of the user interface. Filters allow you to specify a list of objects with static criteria. IntelliSearch allows you to specify complex queries in a simple way using prefixes. Prefixes are used to specify which property you are searching in and have short and long syntaxes. For example, if you want to display all reports with the term *import* in the comment, you would use the search criteria as follows:

```
c:import
```

You can search more than one property, as follows:

```
c:import n:Bi
```

Note

No spaces should be entered before or after the colon (:).

See the Advanced Object Search for more information.

The following table illustrates the available prefixes for reports:

Prefixes	Description
n, name	searches the name property
c, com, comment	searches the comment property
d, desc, description	searches the description property
a, app, application	searches the application property

Deleting Reports

Reports can deleted without having to delete the jobs that generated the report output. You should make sure that all jobs have finished before you delete a report.

Security

Privilege	Description
Report.Create	Create reports
Report.Delete	Delete reports
Report.Edit	Edit reports
Report.View	Access reports

You can grant privileges on two levels, *Access* and *Admin;* a privilege granted on *Admin* level allows the grantee to grant the privilege to other users. These privileges can be granted *system-wide*, per *partition* or *isolation group*.

If you have the security module, which requires the *Module.Security* license key, you have an additional *Security* tab on the report. It allows you to specify which users can access, edit, and delete the report.

- Granting and Revoking System Privileges
- Granting or Revoking Object Privileges

Procedure

Create the report

1. Navigate to *Definitions* > *Reports*.
2. Choose *New* from the context menu.
3. On the *Report* tab, fill a name into the *Name field* and choose a filter name in the query field.
4. On the *Columns* tab choose *Create All Columns* and remove the *LLP Version* column from the list.
5. On the *Selections* tab choose a column and a condition.
6. On the *Sorting* tab, choose the columns you want to sort by and give each a unique sort order.

Generate the report

1. Navigate to *Definitions* > *Job Definitions*
2. Choose *Submit* from the context menu of the job definition *System_ReportRun*.
3. Fill a report name and format into the fields and choose *Submit*.
4. Navigate to *Monitoring* > *Jobs* and locate the job you just submitted.
5. In the lower details pane of the job, expand *Files* and choose *report*.

Example

Create a report for the error jobs in the production environment

1. Navigate to *Definitions* > *Reports*.
2. Choose *New* from the context menu.
3. On the *Report* tab, fill `AllErrorJobs` into the *Name field* and `Job.state.Error` into the query field.
4. On the *Columns* tab choose *Create All Columns* and remove the *LLP Version* column from the list.
5. On the *Selections* tab fill *Remote System* into the `Left Report Column` field, choose *Matches Regex* in the *Operation* field, and enter `PR.*` in the *Value* field of the *Column/Value* area. You have to choose the radio button to fill the *Value* field.
6. On the *Sorting* tab, enter *JobId*.
7. Choose *Save and Close* to leave the edit window.

Generate the report and view its output

1. Navigate to *Definitions* > *Job Definitions*
2. Choose *Submit* from the context menu of the job definition *System_ReportRun*.
3. Fill `AllErrorJobs` *Report* field, choose *HTML* in the *Report Format* field and choose *Submit*.
4. Navigate to *Monitoring* > *Jobs* and locate the job you just submitted.
5. In the lower details pane of the job, expand *Files* and choose *report*.

Basic Notification and Alerting with the Mail Module

Use

The mail module allows you to send emails from within SAP CPS. This module has basic alerting and notification features to alert operators about a job that has reached *Error* or about the publication of a new report, for example. You can have the mails forwarded to mobile phones via SMS, however, you need to use a third party service provider (additional fees may apply). The mail module allows you to easily send standard emails as well as customize the email before it is sent. Two job definitions are shipped with the module:

- System_Mail_Configure - configure the default mail server
- System_Mail_Send - send an email

Note

Use of the Mail module requires the *Module.Mail* key in your license.

Note

The notification and alerting features of the Mail module require you to configure every job chain that is to use these features. If you make extensive use of alerts and/or notifications, Redwood recommends the *Active Alerting* module, as you can configure alerts and notifications centrally. Please see the Different Versions section for more information.

Configuring the Module

Before you can send an email, you need to configure the module to use your mail server and a default `from:` address. The module has a basic SMTP server detection mechanism which only requires the default `From:` email address to retrieve the address of the mail server. If the job was unable to detect your mail server, it will reach *Error*; in this case you will be asked for the `From` address as well as the mail server hostname or IP address.

Note

The SMTP server must accept connections on port `25`.

Attaching Job Files

The mail module allows you to attach output files of jobs to emails. The feature is used in job chains; you create a step with job running *System_Mail_Send* and use the parameters *Relative job* and *Job file* to specify which file to attach to the email. The *Relative job* parameter takes a value in the form of `<step_name>, <job_number>` and the parameter *Job file* takes the short name of the job output file.

Example 1:

```
Relative Job - Step 1, Job 1
Job file - stderr
```

Example 2:

```
Relative Job - Dataload, Job 2
Job file - datarun.txt
```

Advanced Email Customization

The mail module ships with the Mail Job Definition Type which allows you to customize the email before it is sent. See the Customizing Email section of the documentation.

See Also

- Mail Job Definition Type
- Customizing Email

- Configuring Mail and Sending Emails

Configuring Mail and Sending Emails

Use

You configure the Mail Module for a default `SMTP` server and `From` address, these values can be overridden in the *System_Mail_Send* job definition. Once you have configured the Mail Module, you can use the job definition *System_Mail_Send* to send email. The job definition is designed to be included in job chains to email output from other jobs in the job chain and optionally in a step which is only executed when a job has reached status *Error*.

Note

Use of the Mail module requires the *Module.Mail* key in your license.

Note

The notification and alerting features of the Mail module require you to configure every job chain that is to use these features. If you make extensive use of alerts and/or notifications, Redwood recommends the *Active Alerting* module, as you can configure alerts and notifications centrally. Please see the Different Versions section for more information.

SMTP Authentication

SAP CPS supports SMTP authentication when the SMTP username and password are stored in a credential using the smtp credential protocol. SAP CPS looks up the credential with the virtual user, which must be set to *default*, and the SMTP server (endpoint), which must match the value in the */configuration/mail/Server* registry key (the match is case-sensitive). You will have to create the *smtp* credential protocol (the name is case-sensitive) prior to creating the credential.

Procedure

Configuring the mail module with SMTP detection

1. Navigate to *Definitions* > *Job Definitions*.
2. Choose *Submit* from the context menu of the job definition *System_Mail_Configure*.
3. Fill in a valid *From* email address and choose *Submit*.

Sending an email

1. Navigate to *Definitions* > *Job Definitions*.
2. Choose *Submit* from the context menu of the job definition *System_Mail_Send*.
3. Fill in a valid *To* email address and *Subject* and choose *Submit*.

Configuring SMTP Authentication

1. Navigate to *Configuration* > *Credential Protocols*.
2. Choose *New* from the context menu.
3. Fill `smtp` (all lower case) into the *Name* field.

4. Optionally fill the other fields.

5. Choose *Save & Close*.

6. Navigate to *Security > Credentials*.

7. Choose *New* from the context menu.

8. Fill `smtp` into the *Credential Protocol* field, `default` into the virtual user field, the SMTP server hostname into *Endpoint*, and the username/password into the respective fields. Note that the value in the *Endpoint* field must match the value in the */configuration/mail/Server* registry key (case-sensitive).

9. Choose *Save & Close*

Creating a basic notification scenario

1. Navigate to *Definitions > Job Chains*.

2. Choose *New* from the context menu.

3. On the *Job Definition* tab, enter a valid name for the job chain.

4. On the *Job Chain* tab, choose *Add Step* from the context menu of the job chain.

5. Choose *Add Job* from the context menu of the step and specify a job definition, fill in any parameters that are required.

   a. Repeat the previous step(s) for all job definitions that you require to be executed in this job chain.

6. Add another step as described above, for each job for which you want notification, create a separate job for running the job definition *System_Mail_Send*. Fill in the parameters as required.

7. Choose *Save & Close*

Creating a basic alerting scenario

1. Navigate to *Definitions > Job Chains*.

2. Choose *New* from the context menu.

3. On the *Job Definition* tab, enter a valid name for the job chain.

4. On the *Job Chain* tab, choose *Add Step* from the context menu of the job chain.

5. Choose *Add Job* from the context menu of the step and specify a job definition, fill in any parameters that are required.

   a. Repeat the previous step(s) for all job definitions that you require to be executed in this job chain.

6. Add another step as described above, for each job for which you want alerting, create a separate job for running the job definition *System_Mail_Send*. Fill in the parameters as required.

7. Add a *Status Handler* on the job chain, if you want an error in any job to trigger the alert, or on step-level for the required steps.

   a. Select the job chain or the step.

   b. Expand *Default Status Handlers* or *Final Status Handlers*, expand the job statuses for which you want to send an alert.

   c. For each status, select `Goto Step` in the *Action Field* and select the step containing the jobs for alerting in the *Goto Step* field.

   d. On the last step before the *notification* step, expand *Final Status Handlers*, expand the job status *Completed* and select *Mark Completed* in the *Action* field.

   e. This makes sure the alerting step is not executed when no job reached *Error*.

8. Expand *Final Status Handlers* of the alerting step, expand the job status *Completed* and select *Raise Error* in the *Action* field, fill *Message* and *Exit Code* fields appropriately.

9. Choose *Save & Close*

Example

Creating a basic alerting scenario

1. Navigate to *Definitions* > *Job Chains*.

2. Choose *New* from the context menu.

3. On the *Job Definition* tab, enter name `JC_BasciAlert` for the job chain.

4. On the *Job Chain* tab, choose *Add Step* from the context menu of the job chain.

5. Choose *Add Job* from the context menu of the step and specify the job definition *System_Info* in the *Job Definition* field.

6. Choose *Add Step* from the context menu of the job chain.

7. Choose *Add Job* from the context menu of the new step and specify the job definition *System_Mail_Send* in the *Job Definition* field.

8. Select the job chain.

9. Expand *Default Status Handlers*, expand the job status *Error*, *Canceled* and *Killed*.

10. For each status, select `Goto Step` in the *Action Field* and the step *Step 2* in the *Goto Step* field.

11. Expand the job status *Completed* and select *Mark Completed* in the *Action* field.

12. Expand *Final Status Handlers* of the step *Step 2*, expand the job status *Completed* and select *Raise Error* in the *Action* field, fill `An error occurred in the chain` in the *Message* and `123` into the *Exit Code* field.

13. Choose *Save & Close*

Testing the alerting scenario

1. Navigate to *Definitions* > *Job Chains*.

2. Choose *Submit* from the context menu of the job chain *JC_BasciAlert*.

3. On the *Times and Dates* page of the submit wizard, add 3 minutes to the 'Requested Start Time *field value and choose* Next".

4. On the *Scheduling* page, fill `System` into the *Queue* field and choose *Submit*.

5. Navigate to *Monitoring* > *Jobs*.

6. Expand the job chain (you might have to refresh the view), drill down to the first step, choose *Scheduling* > *Cancel* from the context menu of the job of step *Step 1*.

7. Wait until the job chain completes

See Also

- Basic Notification and Alerting with the Mail Module

- Customizing Emails

- Creating Job Chains

Customizing Emails

Use

Using the *System_Mail_Send* you may attach one file and specify email headers, furthermore, the email is in plain text. With the mail job definition type, you can customize the email to meet your needs.

By default, the email is in plain text, this can be overridden by starting the code with a <, in which case HTML is expected. You can specify the content-type in the [Body] section of the email, to set it to XML, for example.

Note

Use of the Mail module requires the *Module.Mail* key in your license.

Note

The notification and alerting features of the Mail module require you to configure every job chain that is to use these features. If you make extensive use of alerts and/or notifications, Redwood recommends the *Active Alerting* module, as you can configure alerts and notifications centrally. Please see the Different Versions section for more information.

Syntax

The mail job definition type has the following sections:

- *[Attachments]* - File attachments are specified here

- *[Headers]* - For email headers

- *[Body]* - Metadata about the body, like content-type and charset.

Blank lines and comments (lines starting with #) are ignored in the sections above.

You can use the following substitution parameters:

- *${jobname}* - The description of the job that sent the email.

- *${jobid}* - The job id of the job that sent the email.

- *${<parameter_name>}* - The value of the parameter of the email job, for example, if the email job has a parameter `Subject` it would be referenced as `${Subject}`. This allows you to use Redwood Expression Language in parameters and use the output of the functions in the mail body.

The following ant expressions can be used to generate the file name of attachments, especially when you use a regex expression:

- *@{name}* - The name of the file to attach.
- *@{index}* - A simple counter.

Attaching Files from the Server

For security reasons, you have to specifically allow the directories in which the files reside on the server to be able to attach them. This is done by setting a key below `configuration/mail/AccessiblePath` in the registry. The registry key does not exist by default and needs to be created. For example, `C:\temp` could be the value of the registry key `configuration/mail/AccessiblePath/tmp`, note that the name `tmp` of the registry key has no significance, it is its value that will be used.

Procedure

Allowing files in `/tmp` to be attached to emails

1. Navigate to *Configuration* > *Registry*
2. Drill-down to `configuration/mail/AccessiblePath`, if the registry key does not exist, create it.
   a. To create the `AccessiblePath` registry key, choose *New* from the context menu of the `mail` registry key and name it `AccessiblePath`.
3. Choose *New* from the context menu of the `AccessiblePath` registry key and give it a name, enter `/tmp` into the value field.
4. Choose *Save & Close*.

See Also

- Mail Job Definition Type
- Basic Notification and Alerting with the Mail Module
- Configuring Mail and Sending Emails

Advanced Notification and Alerting with the Active Monitoring Module

Use

The Active Monitoring Module is used to make sure that errors in specific business processes get acknowledged and fixed by IT staff or escalated when nobody has acknowledged the problem in a specific time-frame. Alerting revolves around the operator message system. When an alert is raised, an operator message is created. This operator message must be replied to in order to acknowledge the alert.

Alerts themselves are children of operator messages, and can be monitored via the operator message monitor. An operator message that has alerts attached to it will have a [+] next to it in the monitor.

There are two ways operators and managers can get informed:

- operator messages
- email/SMS

The key features of the module are:

- Operator message integration - all alerts are raised as operator messages and are visible in the operator message console and the operator messages monitor application.
- Acknowledgment - alerts must be acknowledged by replying to the operator message in order to clear them.
- Alert escalations - unanswered alerts can be escalated through shared escalation pathways (email/SMS).
- Rule based configuration - alerts can be defined as rules separate from the objects that cause the alert to be raised.

Alerts can be defined for the following objects:

- Process servers - any status change. Rules are defined based on a process server name pattern.
- Jobs - any status change. Rules are defined based on a job definition name pattern and the job parameters.

Note

The Active Monitoring Module requires the *Module.Alerting* key to be present in your license.

The alerting system consists of three types of object:

- Alert Sources - object specific rules that are defined for when alerts should be created, see below.
  - Process Server Alert Sources - used when a process server loses the connection to a remote system
  - Job Alert Sources - used when a job reaches an undesired status
  - Ad Hoc Alert Sources - used in job chains to fire alerts with the job definition *System_Alert_Send*
- Alert Escalations - rules detailing who to send the alert to, how long to wait for acknowledgment, and which alert to escalate to should the alert remain unacknowledged.
- Alert Gateways - rules determining how messages are formatted and sent.

An alert is raised by an alert source, creating an operator message that needs to be responded to. The alert source specifies the first alert escalation to use. From then on the alerting escalation system decides how long to wait for acknowledgment, and what the next alert is. While the alert is being escalated, messages are sent via the alert gateways to elicit a response. As soon as a the operator message is replied to, the alert is acknowledged and no further automatic action is taken.

Triggering Alerts

Use

Alert sources define which objects create alerts, under which circumstances and which alert to send. By default, the only response to an alert operator message is *Acknowledge*. This can be changed on the alert source.

There are three types of alert sources:

- job alert sources - defines the situations in which specific jobs raise alerts
- process servers alert sources - defines the situations in which specific process servers raise alerts.
- AdHoc alert sources - used in job chains, for example, with the *System_Alert_Send* job definition.

Each alert source uses specific rules to determine:

- when an alert needs to be raised
- the escalation to use
- the message to send

Job Definition Alert Sources

Use

You can use jobs to generate notifications or alerts when they finish with job alert sources. A job alert source can be created to trigger alerts for multiple jobs with pattern matching. Every Job definition alert source needs at least a job name pattern and a status mapping, or no notifications or alerts will be sent. See Raising Alerts and Notifications from Job Definitions information on how to create a job definition alert source.

Actions

Job definition alert sources support the following actions:

Action	**Description**
Export	Export the job definition alert source into a CAR file
Edit	Edit the job definition alert source
Edit Security	Edit the security of the job definition alert source
Delete	Delete the job definition alert source
Duplicate	Make a copy of the job definition alert source to create a similar one
Expand All	Expand all job definition alert sources in the current filter
New	Create a new job definition alert source
Filter > New Filter	Create a new job definition alert source filter
Filter > Edit Filter	Edit current job definition alert source filter
Filter > Delete	Delete current job definition alert source filter

Criteria

Job definition alert sources raise alerts or notifications when jobs change status.

They match a status on three criteria:

- A pattern matching the name of the job definition, GLOB matching is used, use wildcards (? and *).
- The new status of the job.
- The parameters of the job.

The pattern and the new status must be specified. The parameters are optional. If parameters are specified, then they must match. If no parameters are specified then no additional matching is done. Parameter matches are used to raise alerts or notifications on generic job definitions like SAP_AbapRun. Since all of these jobs have the same job definition name, additional criteria are needed to distinguish specific SAP Systems, programs or variants.

You can use the following match rules for parameters:

- *Exact Insensitive* - The value of the parameter must match, case-insensitive. No wildcards allowed.
- *Exact Sensitive* - The value of the parameter must match, case-sensitive. No wildcards allowed.
- *GLOB Insensitive* - The value of the parameter must match, case-insensitive. GLOB wildcards (? and *) allowed.
- *GLOB Sensitive* - The value of the parameter must match, case-sensitive. GLOB wildcards (? and *) allowed.
- *RegEx Insensitive* - The value of the parameter must match, case-insensitive. Regular Expression pattern.
- *RegEx Sensitive* - The value of the parameter must match, case-sensitive. Regular Expression pattern.

Note

You must use parameter names and not descriptions, when specifying parameters. You can look these up in *Definitions* > *Job definitions* or for built-in job definitions you can see System job definitions and SAP job definitions.

Operator Message and Reply Expressions

The operator message to use is specified as an expression that allows substitution parameters, see the parameters below. The message can be specified at the alert source level, and overridden per status. If no message is specified at any level, the default message is *Acknowledge*.

The reply expression works the same way as the reply expression for the *System_OperatorMessage* system job definitions. This can be specified at the alert source level, and overridden per status. If no reply expression is specified, the default reply expression is *Acknowledge*.

Substitution Parameters

Job definition alert sources support the following substitution parameters:

- ${jobDefinition} - the name of the job definition.
- ${jobDefinitionOwner} - the owner of the job definition.
- ${jobId} - the job id.
- ${topLevelJobId} - the job id of the highest parent.
- ${jobOwner} - the owner of the job.
- ${oldStatus} - the old status of the job.

- ${newStatus} - the new status of the job.
- parameters.name - the value of input parameter 'name' of the job, available in REL only.
- outParameters.name - the value of output parameter 'name' of the job, available in REL only.
- ${topLevelQueue} - the name of the queue of the highest parent job
- ${remoteStatus} - the status in the remote system
- ${returnCode} - the return code of the job
- ${queue} - the queue of the job

Escalation

When nobody has resolved the alert in a timely fashion, you can escalate the alert to another operator, this is done with escalations.

Job definition alert sources use two rules to determine the alert escalation to use:

- An expression to determine a dynamic escalation name. This expression supports ${variable} substitutions and the Redwood Expression Language Syntax syntax for expressions.
- A default alert escalation

The expression is always evaluated first. If the escalation returned by the expression exists, then it is used. If it does not exist, then the default is used. This allows the used escalation to be dynamic.

For example, different escalation paths can be defined depending on the name of the :

- Expression: Error_${processServer}
- Default: Error_ProcessServer
- Statuses: PartiallyRunning, Shutdown, Connecting
- The following alert escalations are defined: Error_Finance, Error_Logistics, Error_ProcessServer

This will raise the following alerts:

- Logistics changes to PartiallyRunning - Error_Logistics
- Finance changes to PartiallyRunning - Error_Finance
- HumanResources changes to Shutdown - Error_ProcessServer

Alert Source Actions

Alert source actions are used to customize the behavior using RedwoodScript.

The following pre-defined objects are available:

Object	Class	Example Code
jcsOutLog	com.redwood.scheduler.infrastructure.logging.api.Logger	jcsOutLog.info("This is an informational message.");
jcsErrLog	com.redwood.scheduler.infrastructure.logging.api.Logger	jcsErrLog.warn("The alert has been canceled.");

jcsAlertSourcePostAlertContext	com.redwood.scheduler.api.scripting.variables.AlertSourcePostAlertActionScriptObject	jcsAlertSourcePostAlertContext.getAlertSource();

Finding Job Definition Alert Sources

You can search for job definition alert sources using filters and the *Search Job Definition Alert Sources* box on the Job Definition Alert Sources tab. This box is known as the *intelliSearch* box and located under your username on the top right-hand side of the user interface. Filters allow you to specify a list of objects with static criteria. IntelliSearch allows you to specify complex queries in a simple way using prefixes. Prefixes are used to specify which property you are searching in and have short and long syntaxes. For example, if you want to display all job definition alert sources with the term *error* in the comment, you would use the search criteria as follows:

```
c:error
```

You can search more than one property, as follows:

```
c:error n:all
```

Note

No spaces should be entered before or after the colon (*:*).

See the Advanced Object Search for more information.

The following table illustrates the available prefixes for job definition alert sources:

Prefixes	**Description**
n, name	searches the name property
c, com, comment	searches the comment property
d, desc, description	searches the description property
a, app, application	searches the application property

Security

Privilege	**Description**
JobDefinitionAlertSource.Create	Create job definition alert sources
JobDefinitionAlertSource.Delete	Delete job definition alert sources
JobDefinitionAlertSource.Edit	Edit job definition alert sources
JobDefinitionAlertSource.View	Access job definition alert sources

You can grant privileges on two levels, *Access* and *Admin;* a privilege granted on *Admin* level allows the grantee to grant the privilege to other users. These privileges can be granted *system-wide*, per *partition* or *isolation group*.

If you have the security module, which requires the *Module.Security* license key, you have an additional *Security* tab on the job definition alert source. It allows you to specify which users can access, edit, and delete the job definition alert source.

- Granting and Revoking System Privileges
- Granting or Revoking Object Privileges

See Also

Raising Alerts and Notifications from Jobs

Raising Alerts and Notifications from Jobs

Use

You can use jobs to generate alerts when they fail with job alert sources. A job alert source can be created to trigger alerts for multiple jobs with pattern matching.

Substitution Parameters

Job definition alert sources support the following substitution parameters:

- ${jobDefinition} - the name of the job definition.
- ${jobDefinitionOwner} - the owner of the job definition.
- ${jobId} - the job id.
- ${topLevelJobId} - the job id of the highest parent.
- ${jobOwner} - the owner of the job.
- ${oldStatus} - the old status of the job.
- ${newStatus} - the new status of the job.
- parameters.name - the value of input parameter 'name' of the job, available in REL only.
- outParameters.name - the value of output parameter 'name' of the job, available in REL only.
- ${topLevelQueue} - the name of the queue of the highest parent job
- ${remoteStatus} - the status in the remote system
- ${returnCode} - the return code of the job
- ${queue} - the queue of the job

Parameters with the `${}` syntax can be entered verbatim, for example:

```
Job with job id ${jobId} failed!
```

As soon as you use REL expressions, like `parameters.name`, you need to use the REL syntax

```
='Parameter Param1: '+parameters.Param1
```

To mix REL and non-REL substitution parameters, proceed as follows:

```
='Job ${jobId} has a parameter Param1 with the following value: '+parameters.Param1
```

Procedure

Creating a job definition alert source

1. Navigate to *Alerting > Job Definitions Alert Sources*.
2. Choose *New* from the context menu.

3. Fill in a name into the *Name* field, into the *Name Pattern* field. Optionally fill the other fields.

4. On the *Statuses* tab choose *New* and select a status in the *Status* drop-down. Optionally fill in a status-specific operator message that overrules the default message.

5. Choose *Save & Close*.

Example

Raise alerts when any `SAP_AbapRun` job enters status `Error, Unknown, Killed or Canceled`.

1. Navigate to *Alerting* > *Job Definitions Alert Sources*.

2. Choose *New* from the context menu.

3. Fill in a name and `SAP_AbapRun*` into the *Name Pattern* field.

4. Fill `Job ${jobOwner}.${jobDefinition} with job id ${jobId} reached status ${newStatus}, has the job been restarted?`.

5. Fill `^Yes|No$` into the *Reply Expression* field, to force users to reply with *Yes* or *No*.

6. On the *Statuses* tab:

   a. Choose *New* and select `Error` in the *Status* drop-down.

   b. Repeat previous step for statuses `Unknown, Killed, and Canceled`.

7. Choose *Save & Close*.

Raise alerts when any `SAP_AbapRun` job where the parameter `SAP_SYSTEMS` has value `ERP` enters status `Error, Unknown, Killed or Canceled`.

For the operator message to work, the job definition needs a parameter `Param1`.

1. Navigate to *Alerting* > *Job Definitions Alert Sources*.

2. Choose *New* from the context menu.

3. Fill in a name and `SAP_AbapRun*` into the *Name Pattern* field.

4. Fill `='Job ${jobOwner}.${jobDefinition} with job id ${jobId} and parameter '+parameters.Param1+'reached status ${newStatus}!'` into the *Operator Message* field.

5. On the *Statuses* tab:

   a. Choose *New* and select `Error` in the *Status* drop-down.

   b. Repeat previous step for statuses `Unknown, Killed, and Canceled`, you can fill a specific operator message for each status.

6. On the *Parameters* tab, choose *New* and fill `SAP_SYSTEMS` into the *Name* field and `ERP` into the *Value* field.

7. Choose *Save & Close*.

A job definition is restarted when it reaches *Error*, the maximum number of restarts is set to 4. You do not want to have to answer an alert for the first restart. Note that `getRestartCount` returns the number of remaining restarts.

```
{
  // Get the alert information
  Alert alert = jcsAlertSourcePostAlertContext.getAlert();
  OperatorMessage om = alert.getOperatorMessage();
  SchedulerEntity se = om.getSenderObject();

  // Do not send emails for the first failure
  if (se instanceof Job)
  {
    Job j = (Job) se;
    Long restartCount = new Long(3L);
    // If RestartJobId is null, then this is the first job.
    if (j.getStatus().getState() == JobStatusState.Final && restartCount.equals(j.getRestartCount()))
    {
      // Reply to the operator message and restart the job
      om.setReply("Acknowledged");
    }
  }
}
```

See Also

Job Definition Alert Sources

Process Server Alert Sources

Use

Process server alert sources raise alerts when process servers change status. Likewise, job alert sources raise alerts when jobs reach a specific status.

They match on two criteria:

- A pattern matching the name of the process server or job.
- The new status of the process server or job.

Alerts will be raised if both *Name* and *Statuses* criteria match. Glob matching is done, so the wildcards `*` and `?` can be used, just like in Microsoft Windows or Unix.

Context-Menu Actions

Process server alert sources support the following actions:

Action	**Description**
Export	Export the process server alert source into a CAR file
Edit	Edit the process server alert source
Edit Security	Edit the security of the process server alert source
Delete	Delete the process server alert source
Duplicate	Make a copy of the process server alert source to create a similar one
Expand All	Expand all process server alert sources in the current filter
New	Create a new process server alert source
Filter > New Filter	Create a new process server alert source filter
Filter > Edit Filter	Edit current process server alert source filter

Filter > Delete	Delete current process server alert source filter

Operator message and reply expressions

The operator message to use is specified as an expression that allows substitution parameters below. The message can be specified at the alert source level, and overridden per status. If no message is specified at any level, the default message is *Acknowledge*.

The reply expression works the same way as the reply expression for the *System_OperatorMessage* job. This can be specified at the alert source level, and overridden per status. If no reply expression is specified, the default reply expression is *Acknowledge*.

Variables

Process server alert sources support the following variables:

- ${processServer} - the name of the process server that changed status.
- ${oldStatus} - the old status of the process server.
- ${newStatus} - the new status of the process server.

Escalation

When nobody has resolved the alert in a timely fashion, you can escalate the alert to another operator. This is done with escalations.

Process server alert sources use two rules to determine the alert escalation to use:

- An expression to determine a dynamic escalation name. This expression supports `${variable}` substitutions and the Redwood Expression Language syntax for expressions.
- A default alert escalation

The expression is always evaluated first. If the escalation returned by the expression exists, then it is used. If it does not exist, then the default is used. This allows the escalation used to be dynamic.

For example, different escalation paths can be defined depending on the new status of a process server:

- Match Name: PS_PRD*
- Expression: PS_PRD_`${newStatus}`
- Default: PS_PRD_Alert
- Statuses: Connecting, PartiallyRunning, Shutdown, Unknown
- The following alert escalations are defined: PS_PRD_Connecting, PS_PRD_Shutdown, PS_PRD_Alert

If the new status of a process server named *PS_PRD_EMEA_DB2*, for example, is:

- *Connecting* - PS_PRD_Connecting will be tried, found and used.
- *Unknown* - PS_PRD_Unknown will be tried, but not found, so PS_PRD_Alert will be used.
- *Shutdown* - PS_PRD_Shutdown will be tried, found and used.
- *Running* - no alert will be raised.

Alert Source Actions

Alert source actions are used to customize the behavior using RedwoodScript.

The following pre-defined objects are available:

Object	Class	Example Code
jcsOutLog	com.redwood.scheduler.infrastructure.logging.api.Logger	jcsOutLog.info("This is an informational message.");
jcsErrLog	com.redwood.scheduler.infrastructure.logging.api.Logger	jcsErrLog.warn("The alert has been canceled.");
jcsAlertSourcePostAlertContext	com.redwood.scheduler.api.scripting.variables.AlertSourcePostAlertActionScriptObject	jcsAlertSourcePostAlertContext.getAlertSource();

Finding Process Server Alert Sources

You can search for process server alert sources using filters and the *Search Process Server Alert Sources* box on the Process Server Alert Sources tab. This box is known as the *intelliSearch* box and located under your username on the top right-hand side of the user interface. Filters allow you to specify a list of objects with static criteria. IntelliSearch allows you to specify complex queries in a simple way using prefixes. Prefixes are used to specify which property you are searching in and have short and long syntaxes. For example, if you want to display all process server alert sources with the term *error* in the comment, you would use the search criteria as follows:

```
c:error
```

You can search more than one property, as follows:

```
c:error n:all
```

Note

No spaces should be entered before or after the colon (*:*).

See the Advanced Object Search for more information.

The following table illustrates the available prefixes for process server alert sources:

Prefixes	Description
n, name	searches the name property
c, com, comment	searches the comment property
d, desc, description	searches the description property
a, app, application	searches the application property

Security

Privilege	Description
ProcessServerAlertSource.Create	Create process server alert sources
ProcessServerAlertSource.Delete	Delete process server alert sources
ProcessServerAlertSource.Edit	Edit process server alert sources

ProcessServerAlertSource.View	Access process server alert sources

You can grant privileges on two levels, *Access* and *Admin;* a privilege granted on *Admin* level allows the grantee to grant the privilege to other users. These privileges can be granted *system-wide*, per *partition* or *isolation group*.

If you have the security module, which requires the *Module.Security* license key, you have an additional *Security* tab on the process server alert source. It allows you to specify which users can access, edit, and delete the process server alert source.

- Granting and Revoking System Privileges
- Granting or Revoking Object Privileges

See Also

Raising Alerts and Notifications from Process Servers

Raising Alerts and Notifications from Process Servers

Use

Process server alert sources raise alerts when process servers change status. Likewise, job alert sources raise alerts when jobs reach a specific status.

They match on two criteria:

- A pattern matching the name of the process server or job.
- The new status of the process server or job.

Alerts will be raised if both *Name* and *Statuses* criteria match. Glob matching is done, so the wildcards * and ? can be used, just like in Microsoft Windows or Unix.

Operator message and reply expressions

The operator message to use is specified as an expression that allows substitution parameters below. The message can be specified at the alert source level, and overridden per status. If no message is specified at any level, the default message is *Acknowledge*.

The reply expression works the same way as the reply expression for the *System_OperatorMessage* job. This can be specified at the alert source level, and overridden per status. If no reply expression is specified, the default reply expression is *Acknowledge*. Note

Variables

Process server alert sources support the following variables:

- ${processServer} - the name of the process server that changed status.
- ${oldStatus} - the old status of the process server.
- ${newStatus} - the new status of the process server.

Escalation

When nobody has resolved the alert in a timely fashion, you can escalate the alert to another operator. This is done with escalations.

Process server alert sources use two rules to determine the alert escalation to use:

- An expression to determine a dynamic escalation name. This expression supports `${variable}` substitutions and the Redwood Expression Language syntax for expressions.
- A default alert escalation

The expression is always evaluated first. If the escalation returned by the expression exists, then it is used. If it does not exist, then the default is used. This allows the escalation used to be dynamic.

For example, different escalation paths can be defined depending on the new status of a process server:

- Match Name: PS_PRD*
- Expression: PS_PRD_`${newStatus}`
- Default: PS_PRD_Alert
- Statuses: Connecting, PartiallyRunning, Shutdown, Unknown
- The following alert escalations are defined: PS_PRD_Connecting, PS_PRD_Shutdown, PS_PRD_Alert

If the new status of a process server named *PS_PRD_EMEA_DB2*, for example, is:

- *Connecting* - PS_PRD_Connecting will be tried, found and used.
- *Unknown* - PS_PRD_Unknown will be tried, but not found, so PS_PRD_Alert will be used.
- *Shutdown* - PS_PRD_Shutdown will be tried, found and used.
- *Running* - no alert will be raised.

Alert Source Actions

Alert source actions are used to customize the behavior using RedwoodScript.

Procedure

1. Navigate to *Alerting* > *Process Server Alert Sources*.
2. Choose *New* from the context menu.
3. Fill in a name into the *Name* field and a pattern into the *Name Pattern* field.
   a. On the *Sources* tab, choose *New* and select a status in the *Status* drop-down.
   b. Repeat the previous step for all required statuses.
4. Choose *Save & Close*.

Example

Raise alerts when any process server enters the `Shutdown`, `PartiallyRunning`, or `Connecting` status.

1. Navigate to *Alerting* > *Process Server Alert Sources*.
2. Choose *New* from the context menu.
3. Fill in a name into the *Name* field and `*` into the *Name Pattern* field.

   a. On the *Sources* tab, choose *New* and select `Shutdown` in the *Status* drop-down.

   b. Repeat the previous step for statuses `PartiallyRunning` and `Connecting`.

4. Choose *Save & Close*.

Raise alerts when process servers whose names start with Finance enter the `PartiallyRunning` or `Connecting` status.

1. Navigate to *Alerting* > *Process Server Alert Sources*.

2. Choose *New* from the context menu.

3. Fill in a name into the *Name* field and `Finance*` into the *Name Pattern* field.

   a. On the *Sources* tab, choose *New* and select `Shutdown` in the *Status* drop-down.

   b. Repeat the previous step for statuses `PartiallyRunning` and `Connecting`.

4. Choose *Save & Close*.

Alert Source Action to ignore process server PR3_PorcessServer, which is undergoing maintenance

```
{
  // Get the alert information
  Alert alert = jcsAlertSourcePostAlertContext.getAlert();
  OperatorMessage om = alert.getOperatorMessage();
  SchedulerEntity se = om.getSenderObject();

  //Check if the process server is PR3
  if ("Process Server PR3_ProcessServer".equals(se.getErrorNameEN()))
    {
       //Optional log entry
       jcsOutLog.info(se.getErrorNameEN()+ " was shut down, ignoring alert.");
       //Reply to operator message
       om.setReply("Acknowledge");
       //Set status of alert to acknowledged
       alert.setStatus(AlertStatus.valueOf("Acknowledged"));
    }
}
```

Note

In the above example, `getErrorNameEN()` is used to retrieve the name of the process server because the method is available on SchedulerEntity. However, since it is mainly used for error messages, it has an object type prefix, which we have to take into consideration when we compare the strings. You could look up the business key of the process server and retrieve the name from the ProcessServer class, instead.

See Also

Process Server Alert Sources

Ad Hoc Alert Sources

Use

Ad hoc alerts are raised by the *System_Alert_Send* job definition or by custom script using RedwoodScript. The job definition or code can be placed into job chains in order to implement customized alerting.

The job takes two parameters:

- The name of the ad hoc alert source
- An optional data parameter that can be used to determine the escalation to use. This is available as a variable in the alert source expressions.

Variables

Ad hoc alert sources support the following variables:

- ${jobId} - the job id of the alerting job.
- ${data} - the value of the data parameter to the alerting job.
- jobId - the job id of the job that raised the ad hoc alert
- topLevelJobId - the job id of the main parent of the chain (outer-most job chain)
- chainJobId - the job id of the parent of the chain (inner-most job chain)

Operator message and reply expressions

The operator message to use is specified as an expression that allows substitution parameters. The message can be specified only at the alert source level. If no message is specified, the default message is *Acknowledge*.

The reply expression works the same way as the reply expression for the *System_OperatorMessage* job, it can be written as a regular expression. This can be specified at the alert source level. If no reply expression is specified, the default reply expression is *Acknowledge*.

Escalation

When nobody has resolved the alert in a timely fashion, you can escalate the alert to another operator, this is done with escalations.

Ad hoc alert sources use two rules to determine the alert escalation to use:

- An expression to determine a dynamic escalation name. This expression supports ${variable} substitutions and the REL syntax for expressions.
- A default alert escalation

The expression is always evaluated first. If the escalation returned by the expression exists, then it is used. If it does not exist, then the default is used. This allows the escalation used to be dynamic.

Alert Source Actions

Alert source actions are used to customize the behavior using RedwoodScript.

The following pre-defined objects are available:

Object	**Class**	**Example Code**
jcsOutLog	com.redwood.scheduler.infrastructure.logging.api.Logger	jcsOutLog.info("This is an informational message.");
jcsErrLog	com.redwood.scheduler.infrastructure.logging.api.Logger	jcsErrLog.warn("The alert has been canceled.");

jcsAlertSourcePostAlertContext	com.redwood.scheduler.api.scripting.variables.AlertSourcePostAlertActionScriptObject	jcsAlertSourcePostAlertContext.getAlertSource();

Context-Menu Actions

Ad hoc alert sources support the following actions:

Action	Description
Export	Export the ad hoc alert source into a CAR file
Edit	Edit the ad hoc alert source
Edit Security	Edit the security of the ad hoc alert source
Delete	Delete the ad hoc alert source
Duplicate	Make a copy of the ad hoc alert source to create a similar one
Expand All	Expand all ad hoc alert sources in the current filter
New	Create a new ad hoc alert source
Filter > New Filter	Create a new ad hoc alert source filter
Filter > Edit Filter	Edit current ad hoc alert source filter
Filter > Delete	Delete current ad hoc alert source filter

Finding Ad Hoc Alert Sources

You can search for ad hoc alert sources using filters and the *Search Ad Hoc Alert Sources* box on the Ad Hoc Alert Sources tab. This box is known as the *intelliSearch* box and located under your username on the top right-hand side of the user interface. Filters allow you to specify a list of objects with static criteria. IntelliSearch allows you to specify complex queries in a simple way using prefixes. Prefixes are used to specify which property you are searching in and have short and long syntaxes. For example, if you want to display all ad hoc alert sources with the term *error* in the comment, you would use the search criteria as follows:

```
c:error
```

You can search more than one property, as follows:

```
c:error n:all
```

Note

No spaces should be entered before or after the colon (*:*).

See the Advanced Object Search for more information.

The following table illustrates the available prefixes for ad hoc alert sources:

Prefixes	Description
n, name	searches the name property
c, com, comment	searches the comment property
d, desc, description	searches the description property
a, app, application	searches the application property

Security

Privilege	Description
AdHocAlertSource.Create	Create ad hoc alert sources
AdHocAlertSource.Delete	Delete ad hoc alert sources
AdHocAlertSource.Edit	Edit ad hoc alert sources
AdHocAlertSource.View	Access ad hoc alert sources

You can grant privileges on two levels, *Access* and *Admin;* a privilege granted on *Admin* level allows the grantee to grant the privilege to other users. These privileges can be granted *system-wide*, per *partition* or *isolation group*.

If you have the security module, which requires the *Module.Security* license key, you have an additional *Security* tab on the ad hoc alert source. It allows you to specify which users can access, edit, and delete the ad hoc alert source.

- Granting and Revoking System Privileges
- Granting or Revoking Object Privileges

See Also

Ad Hoc Alert Source

Sending Ad Hoc Alert Sources

Use

Ad hoc alerts are raised by the *System_Alert_Send* job definition or by custom script using RedwoodScript. The job definition or code can be placed into job chains in order to implement customized alerting.

The job takes two parameters:

- The name of the ad hoc alert source
- An optional data parameter that can be used to determine the escalation to use. This is available as a variable in the alert source expressions.

Variables

Ad hoc alert sources support the following variables:

- ${jobId} - the job id of the alerting job.
- ${data} - the value of the data parameter to the alerting job.
- jobId - the job id of the job that raised the ad hoc alert
- topLevelJobId - the job id of the main parent of the chain (outer-most job chain)
- chainJobId - the job id of the parent of the chain (inner-most job chain)

Operator message and reply expressions

The operator message to use is specified as an expression that allows substitution parameters. The message can be specified only at the alert source level. If no message is specified, the default message is *Acknowledge*.

The reply expression works the same way as the reply expression for the *System_OperatorMessage* job, it can be written as a regular expression. This can be specified at the alert source level. If no reply expression is specified, the default reply expression is *Acknowledge*.

Escalation

When nobody has resolved the alert in a timely fashion, you can escalate the alert to another operator, this is done with escalations.

Ad hoc alert sources use two rules to determine the alert escalation to use:

- An expression to determine a dynamic escalation name. This expression supports ${variable} substitutions and the REL syntax for expressions.
- A default alert escalation

The expression is always evaluated first. If the escalation returned by the expression exists, then it is used. If it does not exist, then the default is used. This allows the escalation used to be dynamic.

Alert Source Actions

Alert source actions are used to customize the behavior using RedwoodScript.

Procedure

Use the `finance` escalation when the *System_Alert_Send* job is run with *DataWareHouseLoaded* as the alert source parameter.

1. Navigate to *Alerting > Ad Hoc Alert Sources*.
2. Choose *New* from the context menu.
3. Fill `DataWareHouseLoaded` into the *Name* and `Finance` into the *Default Alert Escalation* fields.
4. Choose *Save & Close*.

Note

This requires the following alert escalations `Finance`.

Make the escalation dependent on the value of the *data* alert source parameter

1. Navigate to *Alerting > Ad Hoc Alert Sources*.
2. Choose *New* from the context menu.
3. Fill `DataMartReadCompleted` into the *Name*, `DataMart_${data}` into the *Escalation Expression* and `DataMart_Default` into the *Default Alert Escalation* fields.
4. Choose *Save & Close*.

Result

The second procedure uses an escalation expression:

When the *System_Alert_Send* job definition is run with *DataWareHouseReadCompleted* as a parameter, then the escalation run depends on the value of the data parameter:

- data is *USA* - escalation *DataMart_USA* is run
- data is *EMEA* - escalation *DataMart_EMEA* is run
- data is *APJ* - escalation *DataMart_APJ* is run
- data is *India* - escalation *DataMart_Default* is run
- data is *China* - escalation *DataMart_Default* is run

Example

Restart the inner-most chain when an ad hoc alert is sent

```
{
   // Get the alert information
   Alert alert = jcsAlertSourcePostAlertContext.getAlert();
   OperatorMessage om = alert.getOperatorMessage();
   SchedulerEntity se = om.getSenderObject();

  // Restart parent of parent
  Job myJob = (Job) se;
  myJob.getParentJob().getParentJob().resubmit();

}
```

Defining Escalation Routes

Use

By default the only response to an alert operator message is *Acknowledge*. This can be changed on the alert source.

Alert operator messages are created by alert sources. Each alert source uses specific rules to determine:

- when an alert needs to be raised
- the escalation to use
- the message to send
- the valid replies

Alert escalations define where alerts are sent, and how long to wait for an acknowledgment before continuing to the next stage of the escalation.

Each alert escalation has the following fields:

- Address - where the escalation message is to be sent.
- Default Alert Escalation - the next escalation to use.
- Delay amount and units - how long to wait for acknowledgment before the next escalation.

The message is sent to the specified address the moment the escalation is raised. If a default alert escalation is specified, then it will be raised after the delay specified on the current escalation.

Warning

It is possible to create loops in escalation to repeatedly send messages. We recommend you add a delay of at least one minute between escalations to avoid overloading your mail server or clients.

Alert Escalation Actions

Alert escalation actions allow you to customize the way alerts are escalated.

The following pre-defined objects are available:

Object	**Class**	**Example Code**
jcsOutLog	com.redwood.scheduler.infrastructure.logging.api.Logger	jcsOutLog.info("This is an informational message");
jcsErrLog	com.redwood.scheduler.infrastructure.logging.api.Logger	jcsErrLog.warn("Destination cannot be detected, using default.");
jcsAlertEscalationPostAlertContext	com.redwood.scheduler.api.scripting.variables.AlertEscalationPostAlertActionScriptObject	jcsAlertEscalationPostAlertContext.isResend()

Context-Menu Actions

Alert escalations support the following actions:

Action	**Description**
Export	Export the alert escalation into a CAR file
Edit	Edit the alert escalation
Edit Security	Edit the security of the alert escalation
Delete	Delete the alert escalation
Duplicate	Make a copy of the alert escalation to create a similar one
Expand All	Expand all alert escalations in the current filter
New	Create a new alert escalation
Filter > New Filter	Create a new alert escalation filter
Filter > Edit Filter	Edit current alert escalation filter
Filter > Delete	Delete current alert escalation filter

Finding Alert Escalations

You can search for alert escalations using filters and the *Search Alert Escalations* box on the Alert Escalations tab. This box is known as the *intelliSearch* box and located under your username on the top right-hand side of the user interface. Filters allow you to specify a list of objects with static criteria. IntelliSearch allows you to specify complex queries in a simple way using prefixes. Prefixes are used to specify which property you are searching in and have short and long syntaxes. For example, if you want to display all alert escalations with the term *management* in the comment, you would use the search criteria as follows:

```
c:management
```

You can search more than one property, as follows:

```
c:management n:corp
```

Note

No spaces should be entered before or after the colon (:).

See the Advanced Object Search for more information.

The following table illustrates the available prefixes for alert escalations:

Prefixes	Description
n, name	searches the name property
c, com, comment	searches the comment property
d, desc, description	searches the description property
a, app, application	searches the application property

Deleting Alert Escalations

You can only delete alert escalations when no other objects relate to them. For example, if there are job definition alert sources that use the escalation, the escalation cannot be deleted until all job definition alert sources that use it have been modified. You can see all job definition alert sources that relate to the escalation in *Related Objects* in the lower detail pane and on the show page.

The table in related objects contains three columns:

- Type - the type of object with a link to it
- Related Object - the name of the object with a link to it
- Used As - objects can sometimes be used in different roles

Security

Privilege	Description
AlertEscalation.Create	Create alert escalations
AlertEscalation.Delete	Delete alert escalations
AlertEscalation.Edit	Edit alert escalations
AlertEscalation.View	Access alert escalations

You can grant privileges on two levels, *Access* and *Admin;* a privilege granted on *Admin* level allows the grantee to grant the privilege to other users. These privileges can be granted *system-wide*, per *partition* or *isolation group*.

If you have the security module, which requires the *Module.Security* license key, you have an additional *Security* tab on the alert escalation. It allows you to specify which users can access, edit, and delete the alert escalation.

- Granting and Revoking System Privileges
- Granting or Revoking Object Privileges

Procedure

Create a simple alert escalation

1. Navigate to *Alerting > Alert Escalations*.
2. Choose *New* from the context menu.
3. Fill the *Name* field, the *Address* field, and *Delay* field.
4. Choose *Save & Close*.

Example

Simple three stage escalation

1. Navigate to *Alerting > Alert Escalations*.
2. Choose *New* from the context menu.
   a. Fill `StageOne` in the *Name* field, `duty-operator@masalan.com` into the *Address* field, and `15` into the *Delay* field, leave the *Unit* drop-down on *Minutes*.
   b. Repeat the previous two steps to create two more with the data below.
3. Choose *Save & Close*.

- *Name* - `StageTwo`
- *Address* - `backup-operator@company.com`
- *Default* - `StageThree`
- *Delay* - `10` Minutes
- *Name* - `StageThree`
- *Address* - `it-manager@company.com`
- *Default* - `none`
- *Delay* - `ignored`

Three stage escalation with loop

1. Navigate to *Alerting > Alert Escalations*.
2. Choose *New* from the context menu.
   a. Fill `StageOne` in the *Name* field, `duty-operator@masalan.com` into the "Address' field, and `15` into the *Delay* field, leave the *Unit* drop-down on *Minutes*.
   b. Repeat the previous two steps to create two more with the data below.
3. Choose *Save & Close*.

- *Name* - `StageTwo`
- *Address* - `backup-operator@company.com`
- *Default* - `StageThree`
- *Delay* - `10` Minutes

- *Name* - `StageThree`
- *Address* - `it-manager@company.com`
- *Default* - `StageThree`
- *Delay* - `5` Minutes

Result

Procedure 1:

If an alert is raised at 20:00 and escalated to StageOne, the following will happen:

- 20:00 - An email will be sent to duty-operator@company.com. If no response is received after 15 minutes, then it will be escalated to StageTwo.
- 20:15 - An email will be sent to backup-operator@company.com. If no response is received after 10 minutes, then it will be escalated to StageThree.
- 20:25 - An email will be sent to it-manager@company.com.

Procedure 2:

If an alert is raised at 20:00 and escalated to StageOne, the following will happen:

- 20:00 - An email will be sent to duty-operator@company.com. If no response is received after 15 minutes, then it will be escalated to StageTwo.
- 20:15 - An email will be sent to backup-operator@company.com. If no response is received after 10 minutes, then it will be escalated to StageThree.
- 20:25 - An email will be sent to it-manager@company.com. If no response is received after 5 minutes, then it will be escalated to StageThree.
- 20:30 - An email will be sent to it-manager@company.com. If no response is received after 5 minutes, then it will be escalated to StageThree.
- Email will be continue to be sent to it-manager@company.com every 5 minutes until the operator message is acknowledged.

Email Alert Gateways

Use

Alerting gateways specify the addressee as well as the content of the message that is to be sent to the operator. You can have more than one email gateway, if you want to use multiple SMTP servers.

Actions

Email alert gateways support the following actions:

Action	**Description**
Export	Export the email alert gateway into a CAR file
Edit	Edit the email alert gateway

Edit Security	Edit the security of the email alert gateway
Delete	Delete the email alert gateway
Duplicate	Make a copy of the email alert gateway to create a similar one
Expand All	Expand all email alert gateways in the current filter
New	Create a new email alert gateway
Filter > New Filter	Create a new email alert gateway filter
Filter > Edit Filter	Edit current email alert gateway filter
Filter > Delete	Delete current email alert gateway filter

Patterns

Alert sources use patterns rather than names so that alerts can apply to all objects of a particular type and naming conventions can be used to minimize the number of required alerting rules.

Patterns support two wildcards:

- '*' - matches any number of characters
- '?' - matches any single character.

Example:

- "System_*" - matches all jobs starting with System_
  - System_OperatorMessage - MATCHES
  - SystemJob1 - DOES NOT MATCH
- ER? - matches all three letter strings starting with ER
  - ERP - MATCHES
  - ER1 - MATCHES
  - ERP1 - DOES NOT MATCH

Headers

Email gateways also allow you to override email headers. This is handy if you want to adapt the subject or add a reply-to address, for example. There is a special email header that can be used to override the SMTP server, named `X-Redwood-Server`. You can also add other custom email headers that may be required by your email system. You can use substitution parameters in email gateways, this allows you to dynamically set the subject of the email, for example.

The following substitution parameters can be used in headers:

- ${address} - the addressee's address
- ${messageTitle} - the title of the message
- ${subject} - the subject
- ${message} - the message
- ${alertSource} - the alert source that triggered the alert

- ${object} - the object that triggered the alert

Job Files

You can attach job files of the failed job to the email. This allows the operator to troubleshoot the problem without the need to connect to the system and gives other addressees, like managers, information on what went wrong, if they need to know.

Usually, you would send `stderr` which should contain specific error messages. The job output is usually stored in `stdout`, but you can specify any short name of a job file of the job.

Alerting Gateway Actions

Actions allow you to write script custom reactions to alerts in RedwoodScript. Stub code is already available to you and contains predefined objects.

Object	**Class**	**Example Code**
jcsOutLog	com.redwood.scheduler.infrastructure.logging.api.Logger	jcsOutLog.info("This is an informational message");
jcsAlertSourcePostAlertContext	com.redwood.scheduler.api.scripting.variables.AlertSourcePostAlertActionScriptObject	jcsAlertSourcePostAlertContext.getAlertSource();
jcsEmailAlertGatewayPreSendActionContext	com.redwood.scheduler.api.scripting.variables.EmailAlertGatewayPreSendActionScriptObject	jcsEmailAlertGatewayPreSendActionContext.getMessage();
jcsAlertEscalationPostAlertContext	com.redwood.scheduler.api.scripting.variables.AlertEscalationPostAlertActionScriptObject	jcsAlertEscalationPostAlertContext.isResend()

Finding Email Alert Gateways

You can search for email alert gateways using filters and the *Search Email Alert Gateways* box on the Email Alert Gateways tab. This box is known as the *intelliSearch* box and located under your username on the top right-hand side of the user interface. Filters allow you to specify a list of objects with static criteria. IntelliSearch allows you to specify complex queries in a simple way using prefixes. Prefixes are used to specify which property you are searching in and have short and long syntaxes. For example, if you want to display all email alert gateways with the term *management* in the comment, you would use the search criteria as follows:

```
c:management
```

You can search more than one property, as follows:

```
c:management n:corp
```

Note

No spaces should be entered before or after the colon (*:*).

See the Advanced Object Search for more information.

The following table illustrates the available prefixes for email alert gateways:

Prefixes	**Description**

n, name	searches the name property
c, com, comment	searches the comment property
d, desc, description	searches the description property
a, app, application	searches the application property

Deleting Email Alert Gateways

You can only delete gateways when no other objects relate to them. For example, if there are alert escalations that use the gateway, the gateway cannot be deleted until all alert escalations that use it have been modified. You can see all alert escalations that relate to the gateway in *Related Objects* in the lower detail pane and on the show page.

The table in related objects contains three columns:

- Type - the type of object with a link to it
- Related Object - the name of the object with a link to it
- Used As - objects can sometimes be used in different roles

Security

Privilege	**Description**
EmailAlertGateway.Create	Create email alert gateways
EmailAlertGateway.Delete	Delete email alert gateways
EmailAlertGateway.Edit	Edit email alert gateways
EmailAlertGateway.View	Access email alert gateways

You can grant privileges on two levels, *Access* and *Admin;* a privilege granted on *Admin* level allows the grantee to grant the privilege to other users. These privileges can be granted *system-wide*, per *partition* or *isolation group*.

If you have the security module, which requires the *Module.Security* license key, you have an additional *Security* tab on the email alert gateway. It allows you to specify which users can access, edit, and delete the email alert gateway.

- Granting and Revoking System Privileges
- Granting or Revoking Object Privileges

See Also

Creating and Configuring Alert Gateways

Creating and Configuring Email Alert Gateways

Use

Alerting gateways specify the addressee as well as the content of the message that is to be sent to the operator. You can have more than one email gateway, if you want to use multiple SMTP servers. When you configure multiple alert gateways, you must pay attention to the pattern you use as well as the processing order of each email gateway. The most specific address patterns have to have the lowest processing order, or you might have the wrong email gateway being used. For example, an email gateway with an Address Pattern of `*masalan.com` and a processing order of `1` will take precedence over any other email gateway for `masalan.com`, like `de.mas-`

alan.com or us.masalan.com if their processing order is greater than 1. SAP recommends using a catch-all gateway with an extremely high processing order, like 99, to make sure all emails are sent.

Email Alert Gateway Actions

Email alert gateway actions allow you to customize the email and perform specific tasks before it is sent using RedwoodScript. You can, for example, cancel it under specific circumstances.

Procedure

Setting up email

1. Navigate to *Definitions > Job Definitions*.
2. Choose *Submit* from the context menu of *System_Mail_Configure*.
3. Specify only the from address if you do not know the address of the SMTP server. This will attempt to detect the appropriate mail server to use automatically.
4. Check the status of the job in *Monitoring > Job Monitor*.

If the job ends in status *Completed* then the mail server has successfully been detected and stored.

If the job ends in status *Error*, the mail server could not be automatically detected. You should look at the log file for more information. You can re-run the job and specify the mail server to use. You should ask your system or network administrator for the name or IP address of your SMTP mail server.

Creating an email gateway

1. Navigate to *Alerting > Email Gateways*.
2. Choose *New* from the context menu.
3. Fill in the fields *Name*, *Address Pattern* and *Processing Order*.
4. Optionally set custom headers on the *Headers* and specify files to attach on the *Job File Attachments* tab.
5. Choose *Save & Close*.

Example

Creating an email gateway for `masalan.com`

1. Navigate to *Alerting > Email Gateways*.
2. Choose *New* from the context menu.
3. Fill in the fields *Name*, fill `*@masalan.com` into the *Address Pattern* and `99` into *Processing Order*.
4. On the *Headers* tab, choose *New* and fill `From` into the *Name* field and `prd-operators@masalan.com` into the *Value* field.
5. On the *Headers* tab, choose *New* and fill `subject` into the *Name* field and `prd-Job ${topLevelJobId} failed: ${subject}`
6. On the *Job File Attachments* tab choose *New*, enter `stderr` into the *Name* field, choose *New* again and enter `stdout` into the *Name* field.

7. Choose *Save & Close*.

You want to wait until the second run of the job has failed before you send alerting emails, this is done by suppressing the first email and operator message in the alert gateway action:

```
{
  // Get the alert information
  Alert alert = jcsEmailAlertGatewayPreSendActionContext.getAlert();
  OperatorMessage om = alert.getOperatorMessage();
  SchedulerEntity se = om.getSenderObject();

  // Do not send emails for the first failure
  if (se instanceof Job)
  {
    Job j = (Job) se;

    // If RestartJobId is null, then this is the first job.
    if (j.getStatus().getState() == JobStatusState.Final && null == j.getRestartJobId())
    {
      // Suppress sending the email
      jcsEmailAlertGatewayPreSendActionContext.setSend(false);

      // Reply to and delete the operator message
      om.setReply("Acknowledge");
      om.deleteObject();
    }
  }
}
```

See Also

Email Alert Gateways

Setting User Parameters in the Options dialog

Use

The options dialog is accessible from the title-bar and contains the following user parameters that can be changed:

- Language - The language of the client.
- Timezone - The local time zone.
- Locale - The format you want to use for displaying dates and other data that has region-dependent formats can be specified with the locale. You may customize a subset of settings for your preferred locale:
  - Editable Date Field format
  - Long Date format
  - Medium Date format
  - Short Date format
  - Medium Time format
  - Short Time format
  - Thousand Separator

  - Decimal Separator
  - Shorter than 1 day
  - Between 1 and 2 days
  - Longer than 2 days
- Confirm operations - request user confirmation for the following actions:
  - Confirm delete operations
  - Confirm cancel job operations
  - Confirm kill job operations
  - Confirm restart job operations
- Allow edit as XML - A new tab is displayed for every object in its editor window with the XML source of the object definition.
- UI Color # - The color of the upper part of the tool-bar in hex, default is `3366BB`

Locale Format

In the *Locale format* you can set format preferences for date formats.

The following variables can be used:

`y` - format for the year, two letters displays the year of the century, like `11` for `2011`, four letters display the full year

`M` - format for the month, one letter displays the month in a number format (1 for January), three letters display the short three letter form (JAN for January), four letters display the full name of the month.

`d` - format for the day

`h` - format for the hour

`mm` - format for the minute

`ss` - format for the seconds

The Registry

Use

The registry contains the configuration settings used by clients and the system. The registry is stored in the database and not in the Windows Registry.

A registry entry can have a value associated with it, as well as child registry entries.

Note

Registry entries for users should be managed through the options button on the tool-bar. You should not edit or create registry entries unless asked to do so by SAP support.

The registry is part of the *Configuration* section in the navigation bar. A standard registry tree is shipped with SAP CPS and registry entries for users will automatically be added as users log in.

To view a hierarchy of all registry entries, choose *Configuration* > *Registry* in the Navigation bar.

Registry entries support the basic actions of New, Edit, Delete, and Refresh.

Actions

Registry keys support the following actions:

Action	Description
Export	Export the registry key into a CAR file
Edit	Edit the registry key
Edit Security	Edit the security of the registry key
Delete	Delete the registry key
Duplicate	Make a copy of the registry key to create a similar one
Expand All	Expand all registry keys in the current filter
New	Create a new registry key

Security

Privilege	Description
RegistryEntry.Create	Create registry entries
RegistryEntry.Delete	Delete registry entries
RegistryEntry.Edit	Edit registry entries
RegistryEntry.View	Access registry entries
RegistryEntry.CreateChildRegistryEntry	Create a child registry entry.

You can grant privileges on two levels, *Access* and *Admin;* a privilege granted on *Admin* level allows the grantee to grant the privilege to other users. These privileges can be granted *system-wide*, per *partition* or *isolation group*.

If you have the security module, which requires the *Module.Security* license key, you have an additional *Security* tab on the registry entry. It allows you to specify which users can access, edit, and delete the registry entry.

- Granting and Revoking System Privileges
- Granting or Revoking Object Privileges

Procedure

To change a users time zone using the registry

1. Choose *Configuration* > *Registry*.
2. Expand *user* > *<username>* > *ui* > *configuration*.
3. In the detail pane choose *Edit* from the *Actions* menu.
4. Enter the desired *Time zone* in the field Value.

5. Choose *Save and Close* to save the modified registry key.

To delete child registry entries

1. Choose *new* to create a child registry entry.
2. Choose one *ChildRegistryEntry* and choose *delete* to delete a child registry entry.
3. Choose *Save & Close* to save the new registry key and close the edit window.

Registry Entry

- *Name* - The name of the registry entry.
- *Description* - An optional description of the registry entry.
- *Comment* - An optional comment for the registry entry.
- *Value* - Value of this registry entry stored as textual data.
- *Type* - Client-dependent type definition of the value.
- *Modify allowed* - Whether users can modify the value of this registry entry themselves.

Scaling the Scheduling Resources on the Central System

Use

Resources become scarce when the system load is not evenly spread out across the system landscape, especially when high load jobs are submitted without control. SAP CPS comes with several ways to spread the load out across systems. Every SAP AS that is allowed to run jobs controlled by SAP CPS has its process server and queue created as soon as the connection is configured. You can use queues to spread the load over multiple process servers by attaching the process servers that can run the jobs to queues. Using *Resources*, jobs can be directed to specific process servers that serve the queue.

This chapter contains the following topics:

1. Using Process Servers
   a. Creating a Process Server
   b. Configuring a Process Server
   c. Controlling Process Servers with RedwoodScript
2. Using Queues
   a. Creating a Queue
   b. Using Queues to Administer Job Execution
   c. Using Resources to distribute the load
   d. Controlling Queues with RedwoodScript

3. Prevent Simultaneous Execution of Processes
   a. Creating Locks
4. Creating Event-Driven Schedules
   a. Raising Events with Files
   b. Files on AS/400 Raise Events
5. Tuning SAP CPS
6. Balancing the Load
   a. XAL Load Balancing
   b. SAP CPS Load Balancing for SAP

Using Process Servers

Use

Process servers are responsible for running and monitoring jobs. They run jobs in local and remote systems. Process servers host services, which allow jobs of different types to be run. The different types of jobs supported by SAP CPS include:

- SAP ABAP jobs (via XBP) in R/3 and InfoPackages and Process Chains (via BW-SCH) in BI and Mass Activities in IS/U.
- System jobs (locally) for maintenance tasks.
- Message-driven Beans with JXBP process servers
- SAPPI jobs with SAPPI process servers
- Platform process servers*
  - Unix shell scripts*
  - Windows Command scripts*
  - HP OpenVMS scripts *
- Perl scripts*
- JCL scripts for IBM z/OS*
- AS/400 jobs*

* (requires a specific license)

Note

Some job definition types require specific licenses, please check your license.

Process servers are part of the *Environment* section of the navigation bar.

Note

When you create or edit a process server, the required services are added automatically to the process server based on the job definition types or the SAP interfaces you configured, so there is no need to change services on a process server. The only exception is for process servers that are used for file events exclusively, there you do not need to specify a job definition type, however, the *PlatformAgentService* has to be added manually.

1. Creating a Process Server
2. Configuring a Process Server
3. Process Server Parameters
4. Controlling Process Servers with RedwoodScript

Actions

The following actions are available for process servers:

Action	**Description**
Start	Start the process server
Stop	Stop the process server
Monitor related Jobs	Shows a job monitor with all jobs that are have, are and are scheduled to run on the process server
Export	Allows you to export all the settings from this process server; you can import it into another SAP CPS environment. Requires the Import/Export module license key.
Edit	Allows you to edit the configuration of the process server; you will need to restart the process server for the changes to take effect.
Edit Security	Allows you to specify which users or roles may view, edit, delete process servers.
Delete	Allows you to delete the process server, see the *Deleting Process Servers* section below for more information on deleting process servers.
Duplicate	Allows you to make a copy of the process server will all its settings.
Get Support Files	Allows you to download log files and operator messages from the process server, this allows you to quickly inspect logs when something goes wrong.

Retrieving Support Files

When you use the action *Get Support Files* a job is prepared so you can choose which log files to retrieve. You select the files you are interested in and submit the job.

A new tab will open once the job has completed with the output of the job, choose *<job id> stdout.log* and a second tab will open with a link to the job, from there you can access the zip file with all the logs.

SAP Process Servers

There can be one or more process servers for every instance of an SAP System in the landscape, for example one per interface (XBP, XMW, XAL, Pi or JXBP) or one per instance/client. This allows SAP Systems and interfaces to be managed in a granular manner and makes it possible to manage scheduling on both a per-system and a per-interface basis. SAP process servers are created for you when you define an SAP System object in SAP CPS.

Finding Process Servers

You can search for process servers using filters and the *Search Process Servers* box on the Process Servers tab. This box is known as the *intelliSearch* box and located under your username on the top right-hand side of the user interface. Filters allow you to specify a list of objects with static criteria. IntelliSearch allows you to specify complex queries in a simple way using prefixes. Prefixes are used to specify which property you are searching in and have short and long syntaxes. For example, if you want to display all process servers with the term *production* in the comment, you would use the search criteria as follows:

```
c:production
```

You can search more than one property, as follows:

```
c:production n:B1P
```

Note

No spaces should be entered before or after the colon (*:*).

See the Advanced Object Search for more information.

The following table illustrates the available prefixes for process servers:

Prefixes	Description
n, name	searches the name property
c, com, comment	searches the comment property
d, desc, description	searches the description property
a, app, application	searches the application property

Deleting Process Servers

You can only delete process servers when no other objects relate to them. For example, if there are jobs that have run on the process server, the process server cannot be deleted until all jobs that ran on it have been deleted. You can see all jobs that relate to the process server in *Related Objects* in the lower detail pane and on the show page. Besides, you cannot delete process servers that are currently running, you must stop the process server first.

The table in related objects contains three columns:

- Type - the type of object with a link to it
- Related Object - the name of the object with a link to it
- Used As - objects can sometimes be used in different roles, SAP jobs are forced to run on the process server that is connected to the appropriate SAP instance.

See Also

- Process Server Parameters
- Process Server Services
- Job Files

- Creating Platform Process Servers
- Privileges Required to use Process Servers
- Creating an AS/400 Process Server

Creating a Process Server

Use

A Process server is used to schedule, control, and execute your jobs. A process server:

- Provides resources.
- Accepts jobs from a queue via a queue provider.
- Has services that execute jobs.
- Can represent a remote system, for example an SAP or AS/400 system.

Services

A service executes jobs, and creates job files. A service can be started and stopped independently of other services. When a process server is stopped, all services are stopped. When a process server is started, all of its services are also started. Most services are related to job definition types; when you assign a job definition type to a process server, the required service is automatically assigned to the process server.

In some circumstances, you might want to assign a service but no job definition type to a process server, for example, when the process server is only used for file events.

Process servers are part of the Environment section in the navigation bar. The following standard process servers are provided with SAP CPS:

- *System* - a process server that runs system jobs - this process server cannot be deleted.

Actions

Process servers support the following actions:

Action	Description
Start	Start the process server
Stop	Stop the process server
Monitor related Jobs	Opens the job monitor with a filter displaying jobs related to the current process server
Export	Export the process server into a CAR file
Edit	Edit the process server
Edit Security	Edit the security of the process server
Delete	Delete the process server
Duplicate	Make a copy of the process server to create a similar one
Get Support Files	Allows you to retrieve files for troubleshooting purposes; such as process server logs, job logs, operator messages
Expand All	Expand all process servers in the current filter
New	Create a new process server

Filter > New Filter	Create a new process server filter
Filter > Edit Filter	Edit current process server filter
Filter > Delete	Delete current process server filter

It is not possible to delete a process server when it is in status *Running*. You must first stop the process server.

Process server parameters have two values, a *Current* value that is being used by the process server right now and a *Desired* value, which is the one that will be used after a restart of the process server. These will be different if you edit the parameters on a running process server but do not restart it. The *Desired* value reflects this change and makes it clear that a change has been made, but is not yet being used. Before you restart a process server in a production environment, it is recommended to check the *Desired* values of the parameters.

Errors and Operator Messages

Process servers raise operator messages in the central 'Operator Messages' system when they encounter warnings and errors. You should check the operator message log if a process server appears not to be behaving correctly, after changing a process server (or the related SAP System), or if a process server does not start.

Deleting Process Servers

You can only delete process servers when no other objects relate to them. For example, if there are still jobs that ran on the process server, then it cannot be deleted until all jobs have been deleted. You can see all jobs that relate to the process server in the *Related Objects* in the lower detail pane and on the show page.

Security

Privilege	**Description**
ProcessServer.Create	Create process servers
ProcessServer.Delete	Delete process servers
ProcessServer.Edit	Edit process servers
ProcessServer.View	Access process servers
ProcessServer.ViewJobs	View the jobs in the process server
ProcessServer.Control	open and close the process server
ProcessServer.UseForFileEvents	Use the process server for file event jobs
ProcessServer.SubmitOnto	Submit jobs onto the process server

You can grant privileges on two levels, *Access* and *Admin;* a privilege granted on *Admin* level allows the grantee to grant the privilege to other users. These privileges can be granted *system-wide*, per *partition* or *isolation group*.

If you have the security module, which requires the *Module.Security* license key, you have an additional *Security* tab on the process server. It allows you to specify which users can access, edit, and delete the process server.

- Granting and Revoking System Privileges
- Granting or Revoking Object Privileges

Procedure

1. Choose *Environment* > *Process Servers*.
2. Choose *New*.
3. Fill in the details, refer to the Values section below.

4. Choose *Save and Close*.

Values

More information about the various tabs of the process server editor is available in the Values for Creating a Process Server section of the documentation.

Example

1. Choose *Environment* > *Process Servers*.
2. Choose *New* and fill the data below into the corresponding fields on the tabs.
3. Choose *Save & Close* to save the new process server and close the edit window.

ProcessServer Tab

```
Name: Main_SystemServer
Default Job Time Zone: Europe/Berlin
Application: [none]
Description: Process server for the main system.
Comment: Process server for the main system.
```

Parameters Tab

```
Name: RootDirectory
Value: /opt/redwood/data
```

Services Tab

```
Service: JobChainService
 Desired Status: Running
```

```
Service: SystemService
Desired Status: Running
```

Providers Tab

```
Queue: System
Held: unchecked
Execution size:
Inclusive: unchecked
Hold level:
Hold Above CPU Load:
Hold Above Page Rate:
Time Window: [none]
```

Resources Tab

None

Job Definition Types Tab

```
JOBCHAIN
```

```
SYSTEM
```

See Also

- Process Server Parameters
- Process Server Services
- Privileges Required to use Process Servers
- Configuring a Process Server
- Process Server Parameters
- Controlling Process Servers with RedwoodScript

Values for Creating a Process Server

Use

The following tables illustrate the options available on the tabs of the process server editor.

Process Server Tab

Field Name	Description
Name	The name of the process server.
Default Job Time Zone	The time zone to use for jobs on this process server.
Application	The name of the application used to group this object.
Description	The description of the process server.

Comment Tab

Field Name	Description
Comment	A comment about the process server.

Parameters Tab

For an extensive list of parameters, refer to the process server parameters section.

Field Name	Description
Name	The process server parameter name.
Value	The process server parameter value.

In the lower detail pane of a process server, parameters can have different values for every type.

Type	Description
Current	The current value that is being used by the running process server.
Desired	The value that is used the next time that the process server is started.
LastWorking	The previous value that was used by the process server.

Services Tab

- *Service* - The service that the process server supports.
- *Desired Status* - The status of the service: Running, ShutdownNormal, ShutdownImmediate, or ShutdownAbort.

Note

When you create or edit a process server, the required services are added automatically to the process server based on the job definition types or the SAP interfaces you configured, so there is no need to change services on a process server. The only exception is for process servers that are used for file events exclusively, there you do not need to specify a job definition type, however, the *PlatformAgentService* has to be added manually.

Queue Providers Tab

Field	**Description**
Queue	The queue name.
Held	Is this queue closed? (Y or N)
Execution Size	The number of concurrent jobs in the queue.
Inclusive	If set to true, the queue size includes waiting jobs.
Hold Level	Jobs below this priority level do not start.
Time Window	The time window during which the queue releases the jobs.
Time Zone	The time zone to use when the time window is used (and it has no time zone).

Resources

- *Resource* - The resource the process server can provide.

Job Definition Types

- *Job Definition Type* - The job type that the process server can run.

Configuring a Process Server

Use

Configuring a process server enables you to tune the way it handles jobs and resources.

For most customers, the default settings are satisfactory, especially at implementation time.

The configuration of a process server can be done at any time, the new settings require a restart of the process server, to become active.

Note

You configure process servers that connect to SAP systems via the SAP System object in SAP CPS.

Procedure

1. Choose *Environment* > *Process Servers*.
2. From the context menu of the process server, choose *Edit*.

3. Fill in the fields on the different tabs, refer to the Creating Process Servers section for the values.

4. Choose *Save and Close*.

Example

A process server is running low on resources and important financial jobs are not being processed in the expected time frame. Further analysis shows that there are time consuming but low priority jobs running. One queue provider, named *MSLN_PRD_LW*, is held, to prevent low priority jobs to be executed on this process server, other process servers are serving this queue. Another queue, *MSLN_PRD_EM* contains both low and high priority jobs, after analysis the important jobs have a priority of 80, by setting the hold level to 80 the other jobs run on the remaining process servers serving this queue.

1. Choose *Environment* > *Process Servers*.

2. From the context menu of the process server, choose *Edit*.

3. Choose the tab *QueueProviders*

4. Choose *MSLN_PRD_LW* select *Held*

5. Choose *MSLN_PRD_EM* and enter 80 into the field *Hold Level*.

6. Choose *Save and Close*.

The process server needs to be restarted, the best way to do this is to hold all queue providers and wait until all jobs have finished, then restart the process server and open all queue providers. Holding a queue provider only affects one process server.

See Also

- Configuring Load Balancing on Platform Agents
- Configuring Platform Agents
- Configuring Platform Agents on Microsoft Cluster Service
- Configuring Platform Agents on Microsoft Windows
- Privileges Required to use Process Servers
- Creating a Process Server
- Job Files
- Process Server Parameters
- Process Server Services

Process Server Services

Use

Process servers have services to execute specific job definition types or connect to specific external systems. A process server can have one or more services to enable it to carry out these specific tasks. SAP process servers are special, in that you cannot create them manually. You have to create an SAP System, which creates a process server for the SAP System.

The following services are available by default:

- AS400Service - connects to AS/400 systems
- JobChainService - allows execution of job chain jobs
- JXBPService - allows execution of Message-Driven Beans jobs
- MailService - allows creation of custom emails
- PlatformAgentService - connects to Platform agents
- SAPPIService - allows control of communication channels
- SAPR3Service - connects to SAP Systems
- ScriptService - allows execution of RedwoodScript jobs
- System - allows execution of System jobs

Services have a status, which can be one of the following:

- Connecting - the service is attempting to connect the remote server
- Running - the service is operational; for connecting services, this status means that the connection has been established between the remote system and the Central Scheduler
- Shutdown - the process server is shutdown

Note

All services of a scheduler need to have a *Running* status for the process server to have the *Running* status.

Note

When you create or edit a process server, the required services are added automatically to the process server based on the job definition types or the SAP interfaces you configured, so there is no need to change services on a process server. The only exception is for process servers that are used for file events exclusively, there you do not need to specify a job definition type, however, the *PlatformAgentService* has to be added manually.

Mixing Process Server Services

You may mix process server services, however, you must be careful when doing so. You can, for example, add the ScriptService to an SAP process server. You should watch out, however, when you mix services that save their files remotely with services that save their files locally. If you override the default settings of `DataRootDirectory` the resulting value must be valid on all systems involved. For example, if you set `DataRootDirectory` to `F:\Some\Directory` on a platform agent process server and you add the ScriptService to this process server, the directory has to exist on the Central Scheduler (where RedwoodScript jobs are processed) or the jobs will fail. If the Central Scheduler and the platform agent process server are not running on the same platform (Windows, Unix, HP OpenVMS), you should not mix services if you have specified a `DataRootDirectory` parameter on the process server. You should create separate a process server with the ScriptService and put it in the same queue(s) as the platform agent as using the System process server for this is not recommended (you cannot stop it).

The following services store files on the Central Scheduler

- AS400Service - connects to AS/400 systems
- JXBPService - allows execution of Message-Driven Beans jobs
- MailService - allows creation of custom emails
- SAPPIService - allows control of communication channels
- SAPR3Service - connects to SAP Systems
- ScriptService - allows execution of RedwoodScript jobs
- System - allows execution of System jobs

The following service stores files on remote systems

- PlatformAgentService - connects to Platform agents

Security

A service cannot be modified by a user, can be seen by all users and as such has no privileges.

Job Files

Use

Jobs generate output, usually at least one output file and when errors occurred there is also a log file. These files can be plain text files or of another format , like PDF, depending on the job. These files are stored on the process server, SAP and System job output gets stored on the Central Scheduler; Remote agents store output of jobs locally. Sufficient free space must be available on these file systems,

The default output file of jobs is named *stdout*, which refers to the concept of the standard output stream of computer systems. The log file is named *stderr* which refers to the concept of standard error stream of computer systems. Additional output files have the name of the output, if for example in a KSH or CMD job definition you direct output to a file, the filename will be the name of the output.

In the following example, the job file will have the name of the output, `done.txt`.

```
echo done > done.txt
```

The location of the output and log files is configured with the *DataRootDirectory* process server parameter, which can be set for all process servers. The DataRootDirectory also contains process server log files, as well as other data used by file event processes.

Note

The process server parameter can have a trailing slash / or \, however, this is not required.

Under the *DataRootDirectory* a `lis` and `log` directory will be created, the first for the output, which by default will have a `txt` extension, and the latter for log files, which by default will have a `log` extension. The extension of the job files depends on the output and log format.

SAP and System Process Servers

SAP and System process servers do not have the parameter in their parameters list and use the default value, which is `java.io.tmpdir`, and depends on the configuration of your J2ee Application Server, where it can be changed centrally.

In the Windows Standalone Edition, the default `java.io.tmpdir`, is set to `${InstallDir}\j2ee\cluster\server1\temp.`

In SAP NetWeaver, this value is set to the system temp directory by default, which is `%windir%\temp` on Windows by default and `/tmp` on most Unix systems by default.

The *DataRootDirectory* of the *System* process server can be set with the job definition System_SetSystemDataRootDirectory.

The *DataRootDirectory* of SAP process servers just needs to be set on the process server. Previous versions required the process server to have a *System* job definition type, this is no longer the case.

Platform Agents

The default value of *DataRootDirectory* is set to `<InstallDir>/var` by default, this value should be changed to a directory on a large file system.

<InstalLDir> is the installation directory, on Windows the default is `%ProgramFiles%\Redwood\Server` on Unix systems there is no default, it will default to the current working directory.

Note

Previous releases had the *DataRootDirectory* set to `<InstallDir>/<version>/var;` Redwood highly recommends to set the *DataRootDirectory* process server parameter before upgrading if you had not set it previously. If you choose not to, file events can loose their state after the upgrade.

See Also

The Environment of Platform Jobs

Controlling Process Servers with RedwoodScript

Use

When you have a lot of repetitive tasks to perform, it is often easier to use script. RedwoodScript allows you to control process servers, and you can edit all process servers, regardless of the total number, with a few lines of code. The following examples illustrate how you can use RedwoodScript to control process servers. SAP does not warrant that this is the most effective way to solve your problem. For more information on RedwoodScript, see the API documentation.

Example

Starting and stopping Process Servers

```
{
  // get the process server
  ProcessServer prodProcessServer = jcsSession.getProcessServerByName("TR2_ProcessServer");

  //stop the test process server
  prodProcessServer.stop();
  jcsSession.persist();
  //start the production process server
  prodProcessServer.start();
  jcsSession.persist();
}
```

Adding a Process Server to a Queue

The following example can be used to add a process server into a queue and display all queues the process server is currently serving.

```
{
  // get the queue and the process server
  Queue queue = jcsSession.getQueueByName("TR50_Queue");
  ProcessServer proc = jcsSession.getProcessServerByName("TR5_ProcessServer");

  //create a queue provider
  proc.createQueueProvider(queue);

  //save data in the database
  jcsSession.persist();

  //go through all queue providers of the process server
  for (Iterator it = proc.getQueueProviders(); it.hasNext();)
    {
      QueueProvider provider = (QueueProvider) it.next();
      //print the queues
      jcsOut.println(provider.getQueue().getName());
    }
}
```

Adding a Job Definition Type

When you add a job definition type to a process server in the user interface, the process server automatically gets the respective service. This is not the case in RedwoodScript, the service has to be added as well. The process server needs to be shutdown for the code to work.

```
{
  //get the process server, the job definition type as well as the service
  ProcessServer proc = jcsSession.getProcessServerByName("TR5_ProcessServer");
  JobDefinitionType jobDefType = jcsSession.getJobDefinitionTypeByName("KSH");
  Service procservice = jcsSession.getServiceByName("PlatformAgentService");

  //add job definition type and service to process server
  proc.createProcessServerJobDefinitionType(jobDefType);
  proc.createProcessServerService(procservice);

  //add the mandatory process server parameters required by the ''PlatformAgentService'' service.

  ProcessServerParameter RemoteHostName = proc.createProcessServerParameter();
  RemoteHostName.setName("RemoteHostName");
  RemoteHostName.setValue("192.168.1.3");

  ProcessServerParameter SharedSecret = proc.createProcessServerParameter();
  SharedSecret.setName("SharedSecret");
  SharedSecret.setValue("SomeBase64");
  jcsSession.persist();
}
```

Shutdown All Process Servers

```
{
  String query = "select ProcessServer.* from ProcessServer where Name <> 'System'";
  for (Iterator it = jcsSession.executeObjectQuery(query, null); it.hasNext();)
    {
      ProcessServer proc = (ProcessServer) it.next();
      jcsOut.println("Stopping process server "+ proc.getName()+", which has the status "+ proc.getStatus()+"
      proc.stop();
      jcsSession.persist();
    }
}
```

See Also

- Process Server Parameters
- Process Server Services
- Job Files
- Creating Platform Process Servers
- Privileges Required to use Process Servers
- &Cronacle8DocAS400ProcessServers;
- Creating a Process Server
- Configuring a Process Server
- Process Server Parameters

Using Queues

Use

Groups of jobs are managed using queues. Queues distribute jobs to process servers, where the jobs actually run.

You can use queues to limit the number of jobs that can run simultaneously, distribute workload across multiple systems and provide fail-over facilities. One queue can be attached to multiple process servers, in which case SAP CPS decides which process server to run the job at when the job starts. One process server can be attached to multiple queues, in which case it can run jobs from any of those queues.

1. Creating a Queue
2. Using Queues to Administer Job Execution
3. Using Resources to distribute the load
4. Controlling Queues with RedwoodScript

Note

SAP recommends you to specify a time zone on a queue, especially when you tie a time window to it to make sure the correct time is used.

Finding Queues

You can search for queues using filters and the *Search Queues* box on the Queues tab. This box is known as the *intelliSearch* box and located under your username on the top right-hand side of the user interface. Filters allow you to specify a list of objects with static criteria. IntelliSearch allows you to specify complex queries in a simple way using prefixes. Prefixes are used to specify which property you are searching in and have short and long syntaxes. For example, if you want to display all queues with the term *production* in the comment, you would use the search criteria as follows:

```
c:production
```

You can search more than one property, as follows:

```
c:production n:B1I
```

Note

No spaces should be entered before or after the colon (*:*).

See the Advanced Object Search for more information.

The following table illustrates the available prefixes for queues:

Prefixes	**Description**
n, name	searches the name property
c, com, comment	searches the comment property
d, desc, description	searches the description property
a, app, application	searches the application property

Deleting Queues

You can only delete queues when no other objects relate to them. For example, if there are jobs that have run in the queue, the queue cannot be deleted until all jobs that ran in it have been deleted. You can see all jobs that relate to the queue in *Related Objects* in the lower detail pane and on the show page.

The table in related objects contains three columns:

- Type - the type of object with a link to it
- Related Object - the name of the object with a link to it
- Used As - objects can sometimes be used in different roles

See Also

- Central Scheduling System
- RedwoodScript
- Privileges Required to use Queues

Creating a Queue

Use

A queue contains jobs waiting to be processed by one or more process servers. Queues allow:

- Resource limiting - you can limit the number of jobs running on a queue.
- Routing using queue providers - queues can route jobs to one or more process servers.

A queue can be served by more than one process server at the same time. You can link a queue to a process server by creating a *queue provider*. When a job enters the queue it will be executed by an active process server that serves the queue.

A queue can be open or held. While a queue is held, the central server will not start any new jobs in the queue.

A queue is automatically held if:

- There are no process servers associated with the queue.
- The time window on the queue is held.
- All queue providers are held. A queue provider is held if:
  - The queue provider has been put on hold by an operator.
  - The process server associated with the queue provider is stopped.

A queue might appear held to you while it is open to others; this is usually due to the fact that you cannot see any of the running process servers that are serving the queue.

A queue can also be held and released by operators, a queue held by an operator will be in status *ClosedHeld*. If a queue is held, no more jobs will start in that queue.

It is also possible to hold the individual queue providers (for example, to hold a provider linking the queue *CRM* to the process server *CR6*). This allows the queue to continue feeding jobs to other process servers. This is useful when it is necessary to shut down a single process server, but leave the queue feeding others. The process to follow in this case is:

- Hold each queue provider that connected to the process server you want to shut down.
- Wait for all jobs currently running on the process server to reach a final state.
- Shut down the process server.

When starting up, the reverse process is necessary:

- Start up the process server.
- Release each queue provider.

Queues are part of the Environments section in the navigation bar. The following standard queues are provided with SAP CPS:

- *System* - the queue for processing system jobs.

Please note that you cannot change (hold, for example) the *System* queue, so you should create your own for scheduling your *System* jobs.

Note

SAP recommends you to specify a time zone on a queue, especially when you tie a time window to it to make sure the correct time is used.

Actions

Queues support the following actions:

Action	Description
Hold	Holds the queue, this prevents new jobs from starting (changing from state Queued to Running) in the queue. Jobs that are already running will continue to run.
Release	Releases the queue, this releases a queue that is held. This action only works if the queue is in status *ClosedHeld* (which means that it is held by the operator).
Monitor related Jobs	Opens the job monitor with a filter displaying jobs related to the current queue
Export	Export the queue into a CAR file
Edit	Edit the queue
Edit Security	Edit the security of the queue
Delete	Delete the queue
Duplicate	Make a copy of the queue to create a similar one
Expand All	Expand all queues in the current filter
New	Create a new queue
Filter > New Filter	Create a new queue filter
Filter > Edit Filter	Edit current queue filter
Filter > Delete	Delete current queue filter

Note

If a queue and/or a queue provider has/have a time window, it is always wise to specify a time window, especially when the time window has not time zone.

Deleting Queues

You can only delete queues when no other objects relate to them. For example, if there are still jobs that ran in the queue, the queue cannot be deleted until all jobs have been deleted, likewise if job definitions reference the queue as a default queue. You can see all jobs and job definitions that relate to the queue in the *Related Objects* in the lower detail pane and on the show page.

Security

Privilege	Description
Queue.Create	Create queues
Queue.Delete	Delete queues
Queue.Edit	Edit queues
Queue.View	Access queues
Queue.ViewJobs	View the jobs in the queue

Queue.SubmitOnto	Submit jobs onto the queue

You can grant privileges on two levels, *Access* and *Admin;* a privilege granted on *Admin* level allows the grantee to grant the privilege to other users. These privileges can be granted *system-wide*, per *partition* or *isolation group*.

If you have the security module, which requires the *Module.Security* license key, you have an additional *Security* tab on the queue. It allows you to specify which users can access, edit, and delete the queue.

- Granting and Revoking System Privileges
- Granting or Revoking Object Privileges

Procedure

1. Choose *Environment* > *Queues*.
2. From the context menu, choose *New*.
3. Fill in the details on the different tabs, refer to the Values section below.
4. Choose *Save and Close*.

Queue

- *Name* - The name of the queue.
- *Application* - The name of the application used to group this object.
- *Description* - The description of the queue.
- *Comment* - A comment about the queue.
- *Execution Size* - The number of concurrent jobs in the queue.
- *Inclusive* - If set to true, the queue size includes waiting jobs.
- *Hold Level* - Jobs below this priority level will not be started.
- *Time Window* - The time window during which the queue will release the jobs.
- *Time Zone* - The time zone to use for the time window (if specified).
- *Overdue* - Set scripts that are not run within their time window to status OVERDUE

Providers

- *Process Server* - The process server name.
- *Held* - Is this queue provider closed? (Y or N)
- *Execution Size* - The number of concurrent jobs in the queue.
- *Inclusive* - If set to true, the queue size includes waiting jobs.
- *Hold Level* - Jobs below this priority level will not be started.
- *Time Window* - The time window during which the queue provider will release the jobs.
- *Time Zone* - The time zone to use for the time window (if specified).

Example

1. Choose *Environment > Queues* in the Navigation bar.

2. Choose *New* and fill in the values below.

3. Choose *Save & Close* to save the new queue and close the edit window.

Queue tab

```
Name: Process_Reports_Queue
Application: [none]
Description: Process Reports Queue
Comment: This queue is used to process reports.
Execution Size: 10
Inclusive: Checked
Hold Level: 20
Time Window: System_Week_Friday
Time Zone: Europe/Berlin
Overdue: Unchecked
```

Providers tab

```
Process Server: SYSTEM
Held: Unchecked
Execution Size: 20
Inclusive: Checked
Hold Above Page Rate:
Time Window: [none]
Time Zone: [none]
```

See Also

- Using Queues to Administer Job Execution

- Using Resources to distribute the load

- Controlling Queues with RedwoodScript

- Privileges Required to use Queues

Using Queues to Administer Job Execution

Use

Jobs wait in queues until a process server serving the queue and that provides adequate resources is available, and while queue limits are not reached.

Queue limits can be a number of jobs, a time window restricting the queue to be open at night time, for example. Queue providers, by which process servers serve a queue, can also have restrictions.

The following restrictions apply to queues and to queue providers:

- *Execution Size*

- *Hold level*

- *Time window* with *time zone*

The most restrictive setting applies, a queue can have an execution size of 3, if a queue provider serving this same queue has an execution size of 6 this value is never reached. If the time window of a queue provider is always closed during the time window of the queue then the process server may not run any jobs from that queue.

Note

SAP recommends you to specify a time zone on a queue, especially when you tie a time window to it to make sure the correct time is used.

Procedure

To limit the amount of jobs in a queue

1. Choose *Environment* > *Queues*.
2. From the context menu of the queue, choose *Edit*.
3. Fill in a numeric value into the field *Execution Size*.
4. Choose *Save & Close*.

See Also

- Creating a Queue
- Using Resources to distribute the load
- Controlling Queues with RedwoodScript
- Privileges Required to use Queues

Using Resources to distribute the load

Use

Resources are a way of restricting jobs to run on process servers that have certain facilities available. Job definitions can optionally require a resource, when they have, they are forced to run on a process server that offers that resource. Such job definitions will only run on a process server that provides that resource. Process servers may provide one or more resources.

Resources are part of the Definitions section in the navigation bar. There are no standard resources provided with SAP CPS

Resources support the basic actions of New, Edit, Delete, and Refresh.

Actions

Resources support the following actions:

Action	Description
Export	Export the resource into a CAR file
Edit	Edit the resource
Edit Security	Edit the security of the resource

Delete	Delete the resource
Duplicate	Make a copy of the resource to create a similar one
Expand All	Expand all resources in the current filter
New	Create a new resource
Filter > New Filter	Create a new resource filter
Filter > Edit Filter	Edit current resource filter
Filter > Delete	Delete current resource filter

Finding Resources

You can search for resources using filters and the *Search Resources* box on the Resources tab. This box is known as the *intelliSearch* box and located under your username on the top right-hand side of the user interface. Filters allow you to specify a list of objects with static criteria. IntelliSearch allows you to specify complex queries in a simple way using prefixes. Prefixes are used to specify which property you are searching in and have short and long syntaxes. For example, if you want to display all resources with the term *production* in the comment, you would use the search criteria as follows:

```
c:production
```

You can search more than one property, as follows:

```
c:production n:Bi
```

Note

No spaces should be entered before or after the colon (*:*).

See the Advanced Object Search for more information.

The following table illustrates the available prefixes for resources:

Prefixes	**Description**
n, name	searches the name property
c, com, comment	searches the comment property
d, desc, description	searches the description property
a, app, application	searches the application property

Deleting Resources

You can only delete resources when no other objects relate to them. For example, if there are still jobs or job definitions that reference the resource, the resource cannot be deleted until all references have been deleted. You can see all references to the resource in *Related Objects* in the lower detail pane and on the show page. Resources used by process servers are shown in the *Process Server Resources* table.

The table in related objects contains three columns:

- Type - the type of object, like *Job Definition* for a job definition with a link to it
- Related Object - the name of the object with a link to it

- Used As - objects can sometimes be used as different object types

For example, resources that are used in job job definitions, will have the call-reference in *Related Objects*. The type will be *Job Definition*, for example and the *Related Object* will be the name of the job definition.

Note

When a job definition is displayed with a timestamp in the name, this indicates that there still is a job that refers to the resource. You might have changed the job definition, but as long as that job is in the system you will not be able to delete the resource.

Security

Privilege	**Description**
Resource.Create	Create resources
Resource.Delete	Delete resources
Resource.Edit	Edit resources
Resource.View	Access resources

You can grant privileges on two levels, *Access* and *Admin;* a privilege granted on *Admin* level allows the grantee to grant the privilege to other users. These privileges can be granted *system-wide*, per *partition* or *isolation group*.

If you have the security module, which requires the *Module.Security* license key, you have an additional *Security* tab on the resource. It allows you to specify which users can access, edit, and delete the resource.

- Granting and Revoking System Privileges
- Granting or Revoking Object Privileges

Procedure

1. Choose *Definitions* > *Resources*.
2. Choose *Refresh* to see all existing resources.
3. Choose *New* to create a resource.
4. Fill in the details, refer to the section below.
5. Choose *Save and Close* to save the resource.

Resource

- *Name* - The name of the resource.
- *Application* - The name of the application used to group this object.
- *Description* - An optional description of the resource.
- *Comment* - An optional comment about the resource.

Resource Providers

- *Process Server* - The name of the process server that will provide the resource.

Example

1. Choose *Definitions* > *Resources* in the Navigation bar.
2. Choose *New* from the context menu and fill in the values below.
3. Choose the *Resource Providers* tab.
4. Choose the *New* button and select a process server that will provide this resource.
5. Choose *Save and Close* to save the new resource.

```
Name: PRINTER
Description: Printer resource
Comment: Process servers that provide this resource have a printer available.
```

Controlling Queues with RedwoodScript

Use

When you have a lot of repetitive tasks to perform, it is often easier to use script. RedwoodScript allows you to control queues, and you can hold all queues, regardless of the total number, with a few lines of code. The following examples illustrate how you can use RedwoodScript to control queues. Redwood Software does not warrant that this is the most effective way to solve your problem. For more information on RedwoodScript, see the API documentation.

Example

Holding a queue

```
{
  // get the queue
  aQ = jcsSession.getQueueByName("TR2_Queue");
  // check to see if the queue is open
  if (aQ.isOpen())
     {
          //hold the queue
          aQ.hold();
          jcsSession.persist();
     }
}
```

Changing the State of a Queue

This is an example of how you can retrieve the status of a queue.

```
{
  //get the queue
  Queue aQ = jcsSession.getQueueByName("TR2_Queue");

  //check if the queue is open
  if (aQ.isOpen())
     {
         //queue is open
         aQ.hold();
         jcsSession.persist();
      }
  else
```

```
        {
          //queue is held
          aQ.release();
          jcsSession.persist();
        }
  }
```

Holding all Queues

You can hold all queues, except system queue.

```
  {
    for (Iterator it = jcsSession.executeObjectQuery("select Queue.* from Queue where Name <> 'System'",null);
      {
       Queue queue = (Queue) it.next();
       if(queue.isOpen())
        {
          jcsOut.println("Holding queue "+ queue.getName()+", which has the status "+ queue.getStatus()+".");
          queue.hold();
          jcsSession.persist();
        }
      }
  }
```

Adding a Process Server to a Queue

```
  {
    Queue queue = jcsSession.getQueueByName("TR1_Queue");

    ProcessServer proc = jcsSession.getProcessServerByName("TR2_ProcessServer");

    queue.createQueueProvider(proc);

    jcsSession.persist();
  }
```

List all Process Servers Serving a Queue

A queue provider is the link between a process server and a queue, to find all the process servers serving a queue, we need to list the queue providers.

```
  {
  for (Iterator it = queue.getQueueProviders(); it.hasNext();)
     {
        QueueProvider provider = (QueueProvider) it.next();
        out.println(provider.getProcessServer().getName());
     }
  }
```

See Also

- RedwoodScript
- Creating a Queue
- Using Queues to Administer Job Execution
- Using Resources to distribute the load

- Privileges Required to use Queues

Prevent Simultaneous Execution of Processes

Use

Multiple processes can be scheduled to run across the scheduling landscape at any given time. Some of these jobs or job chains have dependencies that must be met and that forbid simultaneous execution. One simple way is to create a job chain that executes the jobs in a specific order, often however, one single job or job chain can only run once or when multiple other jobs or job chains have completed.

To prevent two identical jobs or job chains from running at the same time, locks can be used. A lock is an object that can be attached to a job or job chain and that prevents other jobs or job chains with that same lock from starting as long as a job with that lock is running.

An alternative solution, which is not recommended, would be to use a queue with a queue limit of `1`, so only one job can run in the queue at any given time.

- Creating Locks
- Editing Queues

Creating Locks

Use

Locks define job locks. Job locks prevent simultaneous execution of jobs. When two jobs have the same job lock and one of the two has the status {RUNNING} the other job will not start until the first job has finished.

Locks are part of the Definitions section in the navigation bar. There are no standard locks provided with SAP CPS

Actions

Locks support the following actions:

Action	**Description**
Export	Export the lock into a CAR file
Edit	Edit the lock
Edit Security	Edit the security of the lock
Delete	Delete the lock
Duplicate	Make a copy of the lock to create a similar one
Expand All	Expand all locks in the current filter
New	Create a new lock
Filter > New Filter	Create a new lock filter
Filter > Edit Filter	Edit current lock filter
Filter > Delete	Delete current lock filter

Finding Locks

You can search for locks using filters and the *Search Locks* box on the Locks tab. This box is known as the *intelliSearch* box and located under your username on the top right-hand side of the user interface. Filters allow you to specify a list of objects with static criteria. IntelliSearch allows you to specify complex queries in a simple way using prefixes. Prefixes are used to specify which property you are searching in and have short and long syntaxes. For example, if you want to display all locks with the term *import* in the comment, you would use the search criteria as follows:

```
c:import
```

You can search more than one property, as follows:

```
c:import n:Bi
```

Note

No spaces should be entered before or after the colon (*:*).

See the Advanced Object Search for more information.

The following table illustrates the available prefixes for locks:

Prefixes	Description
n, name	searches the name property
c, com, comment	searches the comment property
d, desc, description	searches the description property
a, app, application	searches the application property

Security

Privilege	Description
Lock.Create	Create locks
Lock.Delete	Delete locks
Lock.Edit	Edit locks
Lock.View	Access locks

You can grant privileges on two levels, *Access* and *Admin;* a privilege granted on *Admin* level allows the grantee to grant the privilege to other users. These privileges can be granted *system-wide*, per *partition* or *isolation group*.

If you have the security module, which requires the *Module.Security* license key, you have an additional *Security* tab on the lock. It allows you to specify which users can access, edit, and delete the lock.

- Granting and Revoking System Privileges
- Granting or Revoking Object Privileges

Procedure

1. Choose *Definitions* > *Locks*.
2. Choose *Refresh* to see all existing locks.
3. Choose *New* to create a lock.

4. Fill in the details, refer to the section below.

5. Choose *Save and Close* to save the lock.

Event Definition

- *Name* - The name of the event definition.
- *Application* - The name of the application used to group this object.
- *Description* - An optional description of the event definition.
- *Comment* - An optional comment of the event definition.

Example

Choose *Definitions* > *Locks* in the Navigation bar. Choose the *New* button and fill in these values:

Job Lock

```
Name: DataWareHouseLoadRun
Description: Data Warehouse load is running
Comment: The data warehouse is being loaded. For data warehouse consistency, this process cannot be run concu
```

Creating Event-Driven Schedules with Events

Use

Events allow the creation of complex dependencies, as they can be raised and cleared by completed jobs or manually. If you have platform agents, you can also create file events which allow files to raise events. A job waiting on events will not start until all events it is waiting for have been raised. This guaranties that a job will not start unless all conditions are met for it to start.

Event definitions define event objects. When an event definition is raised, a new event object is created and associated with the event definition.

Raising an event definition object multiple times will create a queue of event objects of which the first event will be raised and all queued event objects will be pending.

Clearing a raised event definition will clear its currently raised event object and, if there are pending event objects queued for the event definition, raise the next event object and put the event definition back into the raised state. If there are no more pending event objects to be raised, the event definition returns to the cleared state.

Events can be raised and cleared manually, by completed jobs, an SAP System repository object, and the *ProcessServerService* (for file events). Jobs can wait for one or more event objects to be raised before they start. The *Raised by* and *Cleared by* fields will clearly reference which object or user raised or cleared the event.

Event Definitions are part of the Definitions section in the navigation bar. There are no standard event definitions provided with SAP CPS *Raiser and Clearer Comments* can be made compulsory, which will force users who manually raise or clear an event to provide a reason for their actions, the user account that raised/cleared the event will be saved as well. For jobs and job chains the raiser/clearer comments can be defined in the job definition / job chain.

The following substitution parameters are available for raiser comments in job definitions and job chains:

- ${jobid} - The job id of the raiser job

- ${jobdescription} - the raiser job name, as displayed by the job monitor at the time of raising the event.
- ${jobstatus} - The status of the job at the time it raised the event.

The following substitution parameters are available for raiser comments of file events:

- ${CurrentDateTime} - Current date & time
- ${CurrentTimeStamp} - Current time
- ${FileDateTime} - File modification date & time
- ${FileTimeStamp} - File modification time
- ${ServerDateTime} - Central server date & time
- ${ServerTimeStamp} - Central server time
- ${server} - Server name
- ${filename} - file name

The default raiser comment is the following:

Event raised by Job:${jobid} (${jobdescription}) going to status ${jobstatus}.

Actions

Event definitions support the following actions:

Action	Description
Raise	Raise the event definition
Clear	Clear the event definition
Clear All Pending	Clears all pending event definitions
Export	Export the event definition into a CAR file
Edit	Edit the event definition
Edit Security	Edit the security of the event definition
Delete	Delete the event definition
Duplicate	Duplicate the event definition
Expand All	Expand all event definitions in the current filter
New	Create a new event definition
Filter > New Filter	Create a new event definition filter
Filter > Edit Filter	Edit current event definition filter
Filter > Delete	Delete current event definition filter

File Events

Events can also be raised by files on servers with a platform agent or on AS/400 systems.

- Triggering Events with Files
- Files on AS/400 Trigger Events

Finding Event Definitions

You can search for event definitions using filters and the *Search Event Definitions* box on the Event Definitions tab. This box is known as the *intelliSearch* box and located under your username on the top right-hand side of the user interface. Filters allow you to specify a list of objects with static criteria. IntelliSearch allows you to specify complex queries in a simple way using prefixes. Prefixes are used to specify which property you are searching in and have short and long syntaxes. For example, if you want to display all event definitions with the term *import* in the comment, you would use the search criteria as follows:

```
c:import
```

You can search more than one property, as follows:

```
c:import n:Bi
```

Note

No spaces should be entered before or after the colon (*:*).

See the Advanced Object Search for more information.

The following table illustrates the available prefixes for event definitions:

Prefixes	**Description**
n, name	searches the name property
c, com, comment	searches the comment property
d, desc, description	searches the description property
a, app, application	searches the application property

Deleting Event Definitions

You can only delete event definitions when no other objects relate to them. For example, if there are still jobs or job definitions that reference the event (both as wait or raise event), the event definition cannot be deleted until all references have been deleted. You can see all references to the event definition in *Related Objects* in the lower detail pane and on the show page. Furthermore, you cannot delete events which are raised, clear them before you delete them.

Note

Raise events are not displayed in the *Related Objects* table as they these events are displayed as *Raised* in the event definition table.

The table in related objects contains three columns:

- Type - the type of object, like job raise event for for a raise event in a job, with a link to it
- Related Object - the name of the object with a link to it
- Used As - objects can sometimes be used as different object types

For example, event definitions that are used in job chains, will have the call-reference in *Related Objects*. The type will be *Job Definition Raise Event*, for example and the *Related Object* will be the name of the job chain followed by a colon and the name of the event.

Security

Privilege	**Description**
EventDefinition.Create	Create event objects
EventDefinition.View	Access events and event objects
EventDefinition.Raise	Raise events
EventDefinition.Clear	Clear events
EventDefinition.Clear Pending Events	Clear pending events
EventDefinition.Delete Archived Events	Delete archived events
EventDefinition.Edit	Edit event objects
EventDefinition.Delete	Delete event objects

You can grant privileges on two levels, *Access* and *Admin;* a privilege granted on *Admin* level allows the grantee to grant the privilege to other users. These privileges can be granted *system-wide*, per *partition* or *isolation group*.

If you have the security module, which requires the *Module.Security* license key, you have an additional *Security* tab on the event definition. It allows you to specify which users can access, edit and delete the event object.

- Granting and Revoking System Privileges
- Granting or Revoking Object Privileges

Procedure

1. Choose *Definitions* > *Event Definitions*.
2. Choose *Refresh* to see all existing event definitions.
3. Choose *New* to create an event definition.
4. Fill in the details, refer to the section below.
5. Choose *Save and Close* to save the event definition.

Event Definition

Tab Name	**Field Name**	**Description**
Event Definition	Partition	The partition of the event definition.
Event Definition	Name	The name of the event definition.
Event Definition	Application	The name of the application used to group this object.
Event Definition	Description	An optional description of the event definition.
Event Definition	Requires Raiser Comment	Does this event definition require a comment when it gets raised?
Event Definition	Raiser Comment	Comment when it gets raised
Event Definition	Requires Clearer Comment	Does this event definition require a comment when it gets cleared?

Event Definition	Clearer Comment	Comment when it gets cleared
Comment	Comment	An optional comment of the event definition.
File Event Definition	Name	The name of the event definition.
File Event Definition	Description	An optional description of the event definition.
File Event Definition	Enabled	Is this file event enabled? Disabled file events will not trigger events.
File Event Definition	Directory	Directory to await the incoming file
File Event Definition	Pattern	Pattern of the file name, can be explicit file name or contain GLOB (* and ?) wildcards
File Event Definition	Move Directory	Once the file has been picked up, move the file (use of this is recommended)
File Event Definition	Poll Interval	Interval to poll the file system.
File Event Definition	Check if file is in use	Check if the file is in use, the process server will wait until the file has been released before raising the event.
File Event Definition	Process Server	The process server which runs on the server.

Example

1. Choose *Definitions* > *Event Definitions* in the Navigation bar.
2. Choose *New* and fill in these values:

Event Definition Tab

```
Name: DataWareHouseLoaded
Description: Data Warehouse loaded
Comment: The data warehouse has been loaded. Reports that require data from the data warehouse should wait fo
```

Raising Events with Files

Use

File events are special events that are linked to a directory on the host that a platform agent runs on. If they are defined the platform agent will monitor the file system and wait for a file to arrive that matches the file pattern. File events are supported by all UNIX, Microsoft Windows and HP OpenVMS platform agents.

If you define a file event on a file event you point it to a single process server. The files that you want events raised for have to be *local* files on the platform agent system connected to by the process server. Where possible the system refuses to detect networked files. If you want networked files to be detected you need to install a platform agent on the system where the files are local. Files located on a SAN are considered local files. NFS or SMB mounted files are considered networked files.

Note

File events are provided by the platform agent, which requires a `ProcessServerService.OS.limit` license key for UNIX and Microsoft Windows agents, and a `ProcessServerService.VMS.limit` license key for OpenVMS agents.

The file event definition sets the:

- Directory where the file(s) are detected.
- Filename or pattern that the file must have.
- Lock flag; if set then the system checks that the file is not locked or open by any process.
- An optional move directory where the file is moved to after it is detected.
- Overwrite flag; if set then detected files will overwrite existing files in the move directory.
- Poll interval; how often the file event is checked.
- Raise comment; allows a custom comment to accompany the raised event.

Detection

If you do not specify a move directory the files are left in place once detected. If their timestamp is changed or they are deleted and then recreated they will be detected again. If you specify a wildcard an event will be raised for every file that changes or is created. To make this work reliably the platform agent keeps a state file that contains the timestamp for every file.

The HP OpenVMS file system supports versioning of files. To ensure reliable detection all versions of a file will be detected, even if you specify a specific version in the filename attribute.

Wildcards

You can specify GLOB style wildcards for the filename: `*` will match any number of characters, `?` will match a single character. You cannot have wildcard characters in the directory path nor can you detect files in multiple directories; if you require that you need to create a file event definition per directory.

Lock detection

The platform agent can check for locks on the file when the lock attribute is set.

On Microsoft Windows and HP OpenVMS this is done by checking that there are no outstanding file locks. This works quickly and reliably on local files.

On UNIX checking for locks would not work as UNIX does not lock files that are open for read or write; therefore we use the "fuser" or equivalent program to check in the kernel whether any process has the inode of the file open. Keep in mind that the "fuser" method is somewhat expensive to check on some UNIX variations; the overhead varies per operating system. If you anticipate having many files that need detecting on a platform agent it is more efficient to modify your setup so that only a marker file is detected that contains no data, removing the need to check for the file still being open.

Changing the lock test on UNIX

You can set the command used to perform the file-is-locked check for UNIX platform agents using the *FileInUseCommand* process server parameter. Its default value is:

Operating System	**Command**
Sun Solaris	{{/etc/fuser ${Filename} }}
Linux	{{lsof ${Filename} }}
Other UNIX	{{/usr/sbin/fuser ${Filename} }}
OpenVMS	N/A
Microsoft Windows	N/A

You can use your own command but it should behave like either fuser or lsof. The rules that the agent uses to parse the output are:

- For fuser, if it finds at least one line of the format {{ ^${file}:[ ]*[0-9]+ }}then the file is locked. Note that the fuser implementation generally either is silent on an unlocked file (for example on Linux) or it produces (for instance on AIX 5): {{ ^${file}:[ ]* }} so the test on digits after the whitespace is necessary.
- For lsof, if the result value is error and the output is empty, the file is not locked. If the result value is ok (and there is output) the file is locked.

The system determines whether to check the lsof type output by scanning the FileInUseCommand for the string `lsof`.

Move Directory

If your wildcard matches many files it is more efficient to use a *move directory*. This allows the system to move detected files to a new directory, keeping the directory that it needs to watch smaller and thus more efficient. This is also helpful at the OS level: if a file is still in the original directory it has not been detected yet; if it is in the move directory the system has seen the file and the event has been raised (or is in the process of being raised). If the filenames that are detected are not unique over time you should set the overwrite flag, or delete the detected file as part of the actions that happen as part of handling the event. By default the system will not overwrite the file, as the content may be needed by the system. If your files contain data you should keep the overwrite file cleared; the platform agent will then make sure that no data is lost.

The move directory field value can be:

1. A full path including a directory and a filename.
2. Just a directory name, in which case the filename will remain the same.
3. Just a file name in which case the file will remain in the same directory with the new filename. It is your responsibility to ensure that the new filename will not match the file event (or else it will be detected with the new name in the next poll.) Using this option is not as efficient as a move directory that contains a directory.

The directory path that the move directory contains can be a subdirectory of the original directory.

Note

The directory path set as the move directory must exist.

The HP OpenVMS file system supports versioning of files. To ensure that no files are clobbered the file move performed by the agent will not modify the file version of the moved file, and refrain from creating a new file version. If you place a file in the detection directory with a version number that already exists in the move directory the file will not be moved and an error is raised unless you have set the *Overwrite* flag.

The following substitution parameters are available for the *Move Directory* field:

- ${Name} - The base filename plus extension of the detected file.
- ${BaseDirectory} - directory excluding filename and extension.
- ${BaseName} - The base filename of the detected file.
- ${Extension} - The extension of the detected file.
- ${Dot} - A dot if the detected file had an extension, and nothing otherwise.
- ${FileDateTime} - The modification timestamp that the detected file has, in a string format `YYYYMMDDHHMMSS`.

- ${FileTimeStamp} - The modification timestamp that the detected file has, in a numeric format containing the milliseconds since midnight of January 1, 1970, usually referred to as epoch or Unix time.
- ${CurrentDateTime} - The current timestamp, in a string format `YYYYMMDDHHMMSS`.
- ${CurrentTimeStamp} - The current timestamp, in a numeric format containing the milliseconds since midnight of January 1, 1970, usually referred to as epoch or Unix time.
- ${FileDateTime:<format>} and ${CurrentDateTime:<format>} - The date/time is formatted according to <format>, which must be a Java SimpleDateFormat.

Note

${DateTime} and ${TimeStamp} have been deprecated and should not be used anymore, the equivalents are ${FileDateTime} and ${FileTimeStamp}, respectively.

Poll interval

The poll interval defines how many seconds elapse between checks for this file event.

Error handling

When a platform agent encounters a problem with a file that it has detected it will take the following actions:

- A message is logged in the network processor log file.
- An operator message attached to the file event definition is sent to the server.
- The file event scan interval is set to a large value. At the first error it is set to one hour. If the value is already one hour or longer it is doubled until the scan interval is once per day. If the next attempt succeeds the scan interval is reset to the original value.

The scan interval is increased as repeating the attempt after the original scan interval would generate huge amounts of error messages.

If you see file event error messages in the operator message console and have repaired the problem then you should reset the platform agent event timing by stopping and then starting the process server. The error messages that cause this are accompanied by a message stating until when the platform agent will refrain from checking the files for this file event. The current list of possible errors includes:

1. Could not check whether file is in use: {command}
2. File event {name} not raised because the file could not be renamed from {source} to {destination}
3. File event {name} not raised for detected file {source} because {directory} is not an existing directory
4. File event {name} not raised for detected file {source} because the variables in {moveDirectory} could not be substituted
5. File event {name} not raised as the target move file name could not be built: Components {moveDirectory} and {targetFile}

Note

Different file events each have an individual scan interval; when one file event definition errors on a particular system then it does not cause any other file events defined on the same process server to delay.

Procedure

1. Choose *Definitions* > *Event Definitions*.
2. Choose *Refresh* to see all existing event definitions.
3. Choose *New* to create an event definition.
4. Fill in the details, refer to the section below.
5. Choose the *File Event Definition* tab and choose the *New* button.
6. Fill in the details, refer to the section below.
7. Choose *Save and Close* to save the event definition.

Event Definition

Tab Name	Field Name	Description
Event Definition	Name	The name of the event definition.
Event Definition	Application	The name of the application used to group this object.
Event Definition	Description	An optional description of the event definition.
Event Definition	Comment	An optional comment of the event definition.
Event Definition	Requires Raiser Comment	Does this event definition require a comment when it gets raised?
Event Definition	Raiser Comment	Comment when it gets raised
Event Definition	Requires Clearer Comment	Does this event definition require a comment when it gets cleared?
Event Definition	Clearer Comment	Comment when it gets cleared
Comment	Comment	An optional comment of the event definition.
File Event Definition	Name	The name of the file event definition.
File Event Definition	Description	An optional description of the event definition.
File Event Definition	Enabled	Is this file event enabled? Disabled file events will not trigger events.
File Event Definition	Process Server	The process server which runs on the server.
File Event Definition	Directory	Directory to await the incoming file
File Event Definition	Pattern	Pattern of the file name, can be explicit file name or contain GLOB (* and ?) wildcards
File Event Definition	Move Directory	Once the file has been picked up, move the file (use of this is recommended at it has less overhead)
File Event Definition	Overwrite file	Whether any existing files in the Move Directory will be overwritten
File Event Definition	Poll Interval	Interval to poll the file system.
File Event Definition	Check Lock	Check the file lock, the process server will wait until the lock on the file has been released.
File Event Definition	Raise Comment	Comment that will accompany the raised event, can contain substitution variables.

Example

1. Choose *Definitions* > *Event Definitions* in the Navigation bar.
2. Choose the *New* button and fill in the values for the *Event Definition Tab*.
3. On the *File Event Definition* tab choose the *New* button and fill in the values below.

Event Definition Tab

```
Name: DataWareHouseLoaded
Description: Data Warehouse loaded
Comment: The data warehouse has been loaded. Reports that require data from the data warehouse should wait fo
```

File Event Definition Tab

```
Name: DataWareHouseLoaded
Description: Data Warehouse loaded
Comment: The data warehouse has been loaded. Reports that require data from the data warehouse should wait fo
Enabled: <true>
Directory: C:\tmp\
Pattern: done??.txt
Move Directory: C:\done\
Check Lock: <true>
Process Server: WIN-SERV-03
```

Leave the other field with their default values.

Tuning SAP CPS

Use

Tuning allows you to customize the behavior of SAP CPS as well as the lower the pressure on SAP Systems.

Tuning SAP CPS

SAP CPS Can be customized by modifying the three submit frames of the following built in job definitions:

- *System_DeleteJobFiles* - delete job logs and output files of deleted jobs.
- *System_ProcessKeepClauses* - delete old jobs according to keep clauses.
- *System_UpdateJobStatistics* - update runtime statistics of jobs.

By default, these jobs are submitted every 15 minutes, you may increase or decrease this by changing the interval of the submit frame.

More information about submit frames can be found in the user guide.

Tuning SAP connections

SAP RFC connections can be tuned, although the default behavior should be ideal for most, if not all implementations. The RFC library in SAP NetWeaver is limited to 100 concurrent RFC connections by default, this can be changed by setting the `CPIC_MAX_CONV` environment variable, please see the SAP documentation for more information, also see SAP Note 314530 and SAP Note 316877.

By default, SAP CPS will open as many RFC connections as required to process the load of jobs. Ideally, SAP CPS would use two or less. However, heavy load and multiple concurrent jobs can require more RFC connections, in which case these will be opened; they are closed again after a pre-defined idle time. If your SAP System is running slow, and the bottleneck has been identified as a high number of open RFC connections, then you can limit the maximum amount of open RFC connections for XBP, XAL and XMW. Note that this can influence the schedule of jobs controlled by SAP CPS.

There is always one reserved RFC connection that the service uses to connect to the SAP System, additionally, there can be one or more RFC connections for every interface. When you tune the RFC connections, bear this extra RFC connection in mind.

RFC connections are mostly used to create and start jobs, like an *SAP_AbapRun* job, and are closed after an idle time. Import jobs, however, like *SAP_ImportCcmsJobs*, will use an RFC connection until the job finishes and the idle time has been reached. There is also a pool of RFC connections that is used to monitor jobs and retrieve job output as well as log files; by default, the maximum pool size is set to 10 and RFC connections will be opened as required until the maximum value has been reached.

When SAP CPS is not monitoring, starting jobs or retrieving output, there should only be one RFC connection open after the idle time of the other RFC connections.

Tuning JMS Messaging

JMS messaging is used in clusters for communication between cluster nodes. If you do not have a cluster and JMS messaging is enabled, which should not be the case, you should turn it off. To make 100% sure that JMS messaging is enabled, submit the `System_Info` job definition, check the output for `Is Really Clustered: true`. If this is set to `true` and you are sure you do not have a multi-node cluster, you can turn it off by setting the following JVM parameter:

```
-Dcom.redwood.scheduler.cluster.sap.enabled=false
```

Warning

You should only change this when advised to do so by SAP support services.

Procedure

To edit the submit frame of the job System_UpdateJobStatistics

1. Choose *Environment* > *Submit Frames*.

2. Choose *Refresh* to update the list.

3. from the context menu of the `System_UpdateJobStatisticsSubmitFrame` submit frame, choose *Edit*.

4. Choose The *Elements* tab and edit the value after `Every Number of` as well as the unit, if required.

5. Choose *Save & Close*.

To edit the maximum numbers of RFC connections of an SAP System connection

1. Choose *Environment* > *SAP Systems*.

2. Choose the *Refresh* button to update the list.

3. Choose *Edit* from the context menu of the SAP System to edit.

4. Choose The *XBP* tab and edit the value after `Max. number of RFC connections`.

5. Choose *Save & Close*.

Example

1. Choose *Environment > Submit Frames*.

2. Choose *Refresh* to update the list.

3. From the context menu of the `System_UpdateJobStatisticsSubmitFrame` submit frame, choose *Edit*.

4. Choose The *Elements* tab and enter the following value in the `Every Number of` field `1`, for the `Unit` field choose Days.

5. Choose *Save & Close*.

To edit the maximum numbers of RFC connections of an SAP System connection

1. Choose *Environment > SAP Systems*.

2. Choose *Refresh* to update the list.

3. From the context menu, choose *Edit*.

4. Choose The *XBP* tab and enter the following value after `Max. number of RFC connections`, `10`.

5. Choose the *Save & Close* button to save changes and leave the editor.

Result of the Example

System_UpdateJobStatistics

The job definition System_UpdateJobStatistics is run once a day and no longer every 15 minutes. This has an advantage if you do not need up-to-date job statistics. Shortly after the initial deployment it is very handy to monitor job statistics to get an idea of the real load on the system as well as other information.

RFC connections

The maximum number of RFC connections has been set to `10` for XBP only. If you are using XAL and/or XMW the RFC connections can still increase, you may limit those connections in similar way, on the XAL or XMW tab of the SAP System.

Tuning Memory Usage/Garbage Collection

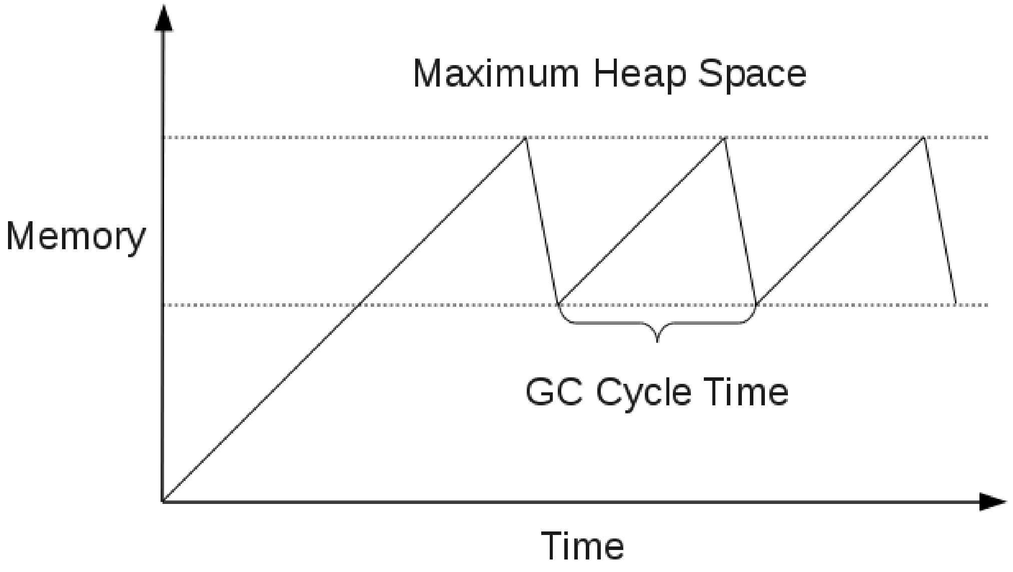
Use

Java uses garbage collection rather than requiring the programmer to free memory explicitly. This means that the amount of memory used will slowly climb until the system detects that it is running low on memory (often triggered by a memory allocation request). At this point the Garbage Collector (GC) runs, and looks around for memory that is allocated but no longer used by the application (called Garbage). It frees some/all of this and then returns the request.

The actual implementation of this varies considerably from Java 1.4 through to Java 7 (1.7), and is considerably more sophisticated than is explained here.

Java 1.4 uses a *stop the world* collector, the application is stopped while GC runs. This can cause pauses (see below).

The two horizontal lines show the minimum required and maximum used heap sizes. If these are close together, then increasing the maximum heap size will be helpful. If they are too far apart, you may get pauses as large amounts of garbage is collected.



Problems and Solutions

If you are getting OutOfMemoryError (OOME) then you need to increase the maximum heap size, simple.

If you are getting long pauses where the application is not responding, GC may be running a long time as it has to do a lot of work. Look at the GC cycle time (see graph above), the amount of memory between the two horizontal lines (or how much churn there is) and the GC log (if you have one) to see if this may be the case. If you are not getting a GC at least once a day, then you may be getting this problem. In this case it may make sense to reduce the maximum amount of memory available, or use GC options to increase the frequency (generally by setting a recommended size for something other than `-Xmx`).

On Java 1.5+ you may be able to enable a concurrent garbage collector that cuts down on pauses. This is not possible on NW 7.00 (NW 2004/NW2004s), as they only run on Java 1.4.

If in doubt, and it is possible to increase the amount of memory, take some measurements (a GC log and some timing measurements for performance), increase the amount of memory and take the measurements again. In general Java benefits from more memory allocated.

Getting a GC log

Add the following to the startup (works with the Sun VM and derivatives):

```
-XX:+PrintGCTimeStamps -XX:+PrintClassHistogram -XX:+PrintGCDetails -verbose:gc -Xloggc:/path/to/gc.log
```

You can check this before changing any NW settings, make sure the path for `–loggc:` is ok, and run:

```
/path/to/java/used/by/nw/bin/java -XX:+PrintGCTimeStamps -XX:+PrintClassHistogram -XX:+PrintGCDetails -verbos
```

If gc.log is created, then the options are well enough supported that you can pass them to NW. Once you have re-configured and restarted NW you can look at the GC log for more information.

Other logs to look for

Generally (it depends on the VM) OOME is logged to `stdout` or `stderr` (and thus should be captured to a file).

- On Redwood Platform: Check if scheduler.xml contains `swallowOutput=true` (in which case the output goes to the web app logger), otherwise check where `stderr` and `stdout` go.
- On NW: Check the default trace file for messages.

Finally, check for `hs_err_pid<xx>.log` files. These should be in `j2ee/cluster/server<n>` (on NW and RP) but are generally written to the current working directory, which may be somewhere else.

Memory in 64 and 32 bit systems

Most platforms now have a 64-bit mode that allows access to more memory. There are some areas to watch out for:

- Some platforms allow you to run a 32-bit VM on a 64-bit CPU architecture, if you need less than the limit that the 32-bit VM supports (generally around 2Gb), use this.
- 64-bit virtual machines use nearly 2x as much memory (as all pointers are now twice the size). So, if you need to go over the limit the 32-bit VM for a given platform supports, you will see a big jump in memory usage when switching to 64-bit.

Newer virtual machines (Java 1.6 and above on most platforms[5,6,7] implement something called *object pointer compression* to alleviate this problem. This allows large heap sizes (>4Gb) without the downside of 64 bit pointers.

On IBM JVM 6, object pointer compression is enabled by default, on Oracle JVM you enable it with the `-XX:+UseCompressedOops` JVM parameter.

Memory errors during startup

In some cases (for example when far too little memory is allocated) you may not be able to start the application server in order to perform some action. In this case you need to tell the application server not to start the application. On SAP NetWeaver you can do this using SAP Notes 914045 – Initially_Not_Started_Apps

Balancing the Load

Use

Load balancing allows you to get the most out of your servers by evenly distributing the load across systems.

There are three types of load balancing for SAP workload that can be used with SAP CPS.

- *CCMS load balancing* - as available with CCMS. This uses a round-robin approach, which is not optimal when the application servers are not equal in size or configuration. No further settings need to be changed in SAP CPS.
- *XAL load balancing* - SAP CPS uses the XAL monitoring data to distribute the load across SAP Systems. XAL data can be up to 5 minutes old.

- *Agent load balancing for SAP* - SAP CPS uses near-real-time monitoring data from a platform agent running on the SAP application server.

Note

XAL load balancing and *Agent load balancing for SAP* require the *Module.LoadBalancing* license key.

Note

Agent load balancing for SAP also requires an additional process server per SAP application server in your *ProcessServerService.External.limit* license key.

For load balancing OS jobs across platform process servers there are two options:

- *job count load balancing* - is the default option if no load factors are defined for the process servers. The system sends new jobs to the process server with the least amount of running jobs.
- *OS metric load balancing* - uses near-real-time monitoring data from the platform agents to decide where to run the job.

Note

OS metric load balancing requires the *Module.LoadBalancing* license key.

Note

It is not possible to use SAP CPS to balance the load of BI process chains across SAP application servers. As far as scheduling is concerned, you can only influence the parent job. You can, however, restart certain child jobs.

More detail on how to set up and use load balancing can be found in:

1. XAL Load Balancing
2. Agent Load Balancing for SAP
3. Configuring Load Balancing on Platform Agents

XAL Load Balancing

Use

SAP CPS by default defers the decision as to where to run an SAP job to the SAP System that the job is executed on. In other words: submitting a job in SAPGUI or SAP CPS will have the same effect.

The CCMS scheduler will use a simple round-robin method to distribute ABAP jobs over the application servers. This can be sub-optimal, as this does not take into account the relative performance and desired profile of the application servers.

For example, let's say that you have two application servers, one slow system with three batch work processes and a fast system with six batch work processes. The round-robin algorithm used by the ABAP scheduler will only start using the 4th, 5th and 6th work process once the three work processes on the slower system are used up.

With SAP CPS, you can use the SAP derived operating system monitoring data from application servers to determine the best application server for the job. SAP CPS obtains this data over the XAL interface and fills the `TARGET_SERVER` parameter for you on the job. To enable this you have to configure the XAL interface on SAP System.

The XAL data is calculated over a 5 minute average. If your batch jobs are short and system load varies much more quickly you can also use Agent Load Balancing for SAP as it can provide very quick update rates.

Load balancing over SAP application servers is configured in the *Maintain Objects* context menu of the SAP System definition in *Environment* > *SAP System*. Obviously, your instance needs at least two or more application servers in order for SAP CPS to be able to send jobs to the appropriate application server. Every application server can have its own load factors. The SAP CPS dispatcher will use the load factors to set the `TARGET_SERVER` parameter.

Monitor tree

The system will create a monitor value containing the number of running jobs for every application server named

```
???/RW_SAP/SAP_ApplicationServers/<app_server>
```

If, for instance, one system should handle three times more jobs than the others, you set the multiplier to *1* on it and *3* on the others. The threshold defines when no more jobs should be allocated to the application server: the application server will be considered overloaded when the threshold equals `jobs * load factor`. An additional threshold field, named *Load Threshold* on the *SAP Application Server* tab, allows you to specify when the entire SAP System should be considered overloaded by SAP CPS, this can be set on the process server as well as on individual application servers and once reached, it will set the process server of the SAP System to overloaded. This allows integration into the Active Monitoring Module.

Note

As an SAP System also has a related process server, you might think that you could use the *Load Factors* tab on the `<SID>_ProcessServer` window to influence load balancing. This is correct, but it would decide on which *SAP System* the job would run, not on which SAP application server *within* a particular SAP instance. Therefore, this is probably not what you intend to do.

Note

It is not possible to use SAP CPS to balance the load BI of process chains across SAP application servers. As far as scheduling is concerned, you can only influence the parent job. You can, however, restart certain child jobs.

Prerequisites

- An SAP System with XAL connection data, see XAL interface on SAP System for more information.

Procedure

Create a load balancing scenario where one application server can handle twice as many jobs as the other one

1. Navigate to *Environment* > *SAP Systems*.
2. Choose *Maintain Objects* from the context menu of the SAP System.
3. Expand *Application Servers* and choose the stronger application server, fill *Strong Application Server* into the *Description* field.
4. On the *Load Factors* tab, choose *New* and fill *XAL Data* into the description field.
5. In the *Multiplier* field, enter *1*.

6. In the *Threshold* field enter the maximum amount of concurrent jobs multiplied by the *multiplier*.
7. In the *Monitor Value* field, enter the name of the application server and choose *Save*.
8. Choose the weaker application server, fill *Weak Application Server* into the *Description* field.
9. On the *Load Factors* tab, choose *New* and fill *XAL Data* into the description field.
10. In the *Multiplier* field, enter *2*.
11. In the *Threshold* field enter the maximum amount of concurrent jobs multiplied by the *multiplier*. So in this case, if you want to allow `5` concurrent SAP jobs enter `10` which is `5 * 2`.
12. In the *Monitor Value* field, enter the name of the application server.
13. Choose *Save & Close*.

To disable an application server, so that SAP CPS no longer sends workload to it

1. Navigate to *Environment* > *SAP Systems*.
2. Choose *Maintain Objects* from the context menu of the SAP System.
3. Expand *Application Servers* and choose the application server to disable.
4. Check *Enabled* if the box is checked so that the checkbox becomes empty.
5. Choose *Save & Close*.

To force the process server to Overloaded when an application server is overloaded

1. Navigate to *Environment* > *SAP Systems*.
2. Choose *Maintain Objects* from the context menu of the SAP System.
3. Expand *Application Servers* and choose the application server.
4. Fill the maximum amount of jobs this application server can handle into the *Load Threshold* field. Note that no other application servers of this instance will receive additional workload until this application server can accept new workload.
5. Choose *Save & Close*.

Values

TAB	Field	Description
SAP Application Server	Name	The name of the application server
SAP Application Server	Description	An optional description
SAP Application Server	Load Threshold	The load, *jobs * load factor*, at which the process server should go into overloaded in SAP CPS, should be higher than the Threshold.
SAP Application Server	Active	Is this application server active
SAP Application Server	Enabled	Is this application server enabled
Comment	Comment	An optional comment
Load Factors	Description	An optional description

Load Factors	Multiplier	The load factor multiplier by which the monitor value will be multiplied. The higher the value, the less likely it is for jobs to get allocated to this instance
Load Factors	Threshold	The load (`jobs * multiplier`) at which no more jobs are allocated to the application server. Must be lower than *Load Threshold* on the *SAP Application Server* tab
Load Factors	Monitor Value	The monitor value to use, it is the same as the application server name
Providers	SAP Application Server Group	The application server group(s) the application server belongs to

SAP CPS Load Balancing for SAP

Use

Agent Load Balancing for SAP, also known as *SAP CPS Load Balancing for SAP*, further improves on XAL load balancing by improving the accuracy of the performance metrics on which the scheduler bases its decisions.

The agent load balancing method combines metrics gathered by platform agents with the SAP System information that the system keeps. When making the scheduling decision the load factors set for the application servers no longer refer to XAL data but to data originating from platform agents.

To make this approach possible you need to install a platform agent on all application servers that are part of an SAP instance. If you schedule data for multiple SAP Systems, you can make this decision on a per-instance basis.

You may require this platform agent anyway if you want to schedule OS tasks on those systems.

Note

Installing a platform agent on an SAP application server requires an additional process server in your *ProcessServerService.External.limit* license key, like any other platform agent.

You allocate jobs to application servers based on the current CPU usage of the system where that application server runs by specifying the *CPUBusy* monitor leaf node of the corresponding platform agent in the *Monitor Value* field. The monitor leaf node for memory usage is named *PageRate*. Make sure that you are specifying the monitor leaf node for the correct process server.

The monitor leaf node is displayed in the *Monitor Value* field as follows:

Monitor Value	**Path**
<monitor_leaf_node>	/System/ProcessServer/<process_server>/Performance}}

Note

The monitor node must contain the name of the platform process server, not the name of the process server of the SAP System, usually named *<SID>_ProcessServer*. When installing a platform agent on an SAP application server, SAP recommends to name the process server *<SID>_<application_server>_Agent*.

Note

It is not possible to use SAP CPS to balance the load BI of process chains across SAP application servers. As far as scheduling is concerned, you can only influence the parent job. You can, however, restart certain child jobs.

Prerequisites

- A platform agent on the nodes of the application servers.

Procedure

Create a load balancing scenario where the ABAP workload should be distributed according to CPU load.

1. Navigate to *Environment > SAP Systems*
2. Choose *Maintain Objects* from the context menu of the SAP System
3. Expand *Application Servers* and choose the stronger application server. Enter `Strong Application Server` into the *Description* field.
4. On the *Load Factors* tab, choose *New* and enter *CPU* into the description field.
5. In the *Multiplier* field, enter `1`.
6. In the *Monitor Value* field, choose the *CPUBusy* monitor for the path */System/ProcessServer/<SID>_<AppServer>_Agent/Performance*.
7. Choose *Save*.
8. Choose the weaker application server, fill *Weak Application Server* into the *Description* field.
9. On the *Load Factors* tab, choose *New* and fill *CPU* into the description field.
10. In the *Multiplier* field, enter `1`.
11. In the *Monitor Value* field, choose the *CPUBusy* monitor for the path */System/ProcessServer/<SID>_<AppServer>_Agent/Performance*.
12. Choose *Save & Close*.

High Availability

Use

The very concept of a central job scheduler means that it is likely that it is important that it is operating, as when it is not then background jobs on many computer systems are affected. This section explains how the software operates, how it can be set up so that it is highly available and how the software can inter-operate with third party products that ensure high availability.

High availability depends on a number of factors, some of which are common to all components (like host availability) and some of which are component specific (like how much processing can continue during a network failure). The factors common to all components include highly available infrastructure and eliminating single points of failure. Those that are component specific include how particular failures (host, node, network and software) are handled by individual components.

Factors common to all components

Many high availability factors are common to all components. The SAP CPS architecture has been designed to isolate the effect of many types of failure to the components dependent on the failed resource. These effects can be reduced or eliminated by implementing highly available solutions. For example:

- Host failures will only affect the components on that host. Using Active-Active platform agents can ensure that processing occurs without interruption.
- Network failures only affect components communicating over that link. Processing can continue on both sides of the link, and in many cases processing can complete and report back results when the network returns.

Single Points of failure (SPOF)

A single point of failure (SPOF) is a resource whose failure will cause a larger system failure. It is important to be aware of potential SPOFs when designing a highly available system. The SPOFs in a system depend not only on the architectures of the products involved, but how they are deployed and the hardware dependencies involved. For example a disk drive may be a point of failure, but not if it is provided by a Storage Area Network (SAN).

When analyzing a system for SPOFs it is important to consider both the impact of the SPOF and the cost of eliminating it. In some cases it may be more expensive to remove an SPOF than it is to deal with its potential down time. For example satellite links are often the only connectivity to remote areas. In these cases it is important to raise awareness of these failure points, and to have contingency plans for operation during failures

SAP CPS can be configured to operate without any SPOF if the high availability features are configured correctly, and it is run on appropriate infrastructure. This includes high availability for:

- the central server, including the application server.
- the database .
- the remote systems being managed.

Network dependencies

High availability setups should do their best to reduce dependencies on network resources, or where this is not possible, to ensure that the network resources themselves are highly available. The most important of these are the Domain Name Servers (DNS) and the Network Time Protocol (NTP) servers. If DNS or NTP are in use, then multiple servers should be available.

Another major area where network dependencies should be avoided is network file systems like the UNIX Network File System (NFS) and Windows file shares (using Server Message Block or SMB). Such file systems are not sufficiently fast or reliable for many scenarios, including:

- Storing the software itself. Software should always be deployed locally.
- Storing data intended to be stored on a local disk.

Data Storage

Each component of SAP CPS has a 'data root directory' where important files like tracing, persistent state and job output are stored. It is important that this directory is on a local disk or SAN (not on a network file system) and that it is appropriately sized. Similarly, the tables in the database should have sufficient space to grow, particularly when significant amounts of processing may occur: overnight, over the weekend, or during period end processing.

All components of SAP CPS have an active tracing facility that will log errors (particularly fatal errors) as they are running. In general these trace files do not take up much space (10-100Mb), but when debugging is activated, or a catastrophic failure occurs they can grow more quickly.

Adequate sizing

Adequate sizing is an important part of any high availability solution, as many outages are caused by sizing problems. The most common are:

- Disks, logical volumes or file systems filling up. These prevent data from being written and will eventually stop all processing until the condition is corrected.
- Correctly sized primary and backup servers. If insufficient CPU or memory is allocated and machines are overloaded they can appear to be 'down' when they are in fact very busy.

Testing

High availability setups should be tested regularly, or at least when the HA or software configuration changes. Without testing small errors like typing ".con" instead of ".com" in a host name may not be picked up until fail-over occurs, in which case a small, easily correctable error could cost many hours (or thousands of dollars).

High availability of the central application server

As the Central Scheduler server runs in a Java Enterprise Edition (JEE) application server stack, availability of this stack is paramount. For this reason the SAP CPS

can run in multiple nodes if the application server is configured to do so. This is often called 'clustered' or 'multi-node' mode. When this is the case then:

- Users are load balanced across all the nodes by the JEE Application server, so users can end up on any node.
- All nodes cache read data for improved performance, and participate in cache coherence using JMS.
- One node (the master) performs certain time-critical operations. If this node fails, one of the slave nodes will automatically take over this functionality.

The time critical operations are performed on a single node as this is more accurate (global locks can take 10ms or more to acquire), and allows the use of both aggressive caching and fast (nanosecond) local locking. These significantly boost performance in all scenarios (single and multi-node), and provides faster, more reliable fail-over.

Application server setups

There are three classes of application server setup:

- single node, single host - a single application server node is running the software, on a single host.
- multiple node, single host - multiple application server nodes run the software, but still on on a single host.
- multiple node, multiple host - multiple application server nodes run the software, on multiple hosts.

The "single node, single host" and "multiple node, single host" setups both a separate host fail-over mechanism to handle the host itself failing (and thus all nodes failing). This is possible, but not required in a "multiple-node, multiple host" setup.

The advantages and disadvantages of the different setups are:

Setup	Advantages	Disadvantages
Single node, Single host	Simple set up. Fits in with existing host based fail-over mechanisms	Relies entirely on host-based fail-over. Most efficient use of memory, network and CPU.
Multiple node, Single host	Fits in with existing host based fail-over mechanisms	More complex set up. Some duplication of software and memory usage. Increased CPU usage
Multiple node, Multiple host	No host level high availability scenario needed	Most complex set up. May require an external load balancer. Some duplication of software and memory usage. Increased CPU and network usage.

The multiple mode setups may allow some reconfiguration of the application server (for example JVM parameters) while the application is running, depending on the application server vendor. This does include patching or upgrading the SAP CPS while it is running.

Performance in multiple node setups

Workload in a multiple node setup is not homogeneous, at a given point in time some nodes may be doing very little while others are processing large, complex tasks. This can result in different performance profiles on different nodes.

Multiple node setups require more hardware resources, particularly memory. In order to operate the SAP CPS software, its data (including cache), the application server and operating system all need to be in memory. Having a single node per host leaves the most memory available for data and cache, as there is only one copy of the other items (application server, SAP CPS software and OS). With multiple nodes per host additional copies of the application server, SAP CPS software are required. Hence the nodes will end up having to share the remaining memory for their data and cache. This means that there is less memory for each node's cache, and that some items may be cached twice (once on each node). This affects performance (more database accesses) and is less efficient (due to the duplication of the application server, SAP CPS software and cache).

Node & Host failure

Both node and host failures are dealt with by the same mechanism. If a node or host fails then other nodes will continue processing their current tasks, depending on the type of node that failed (master or slave), other activities may take place:

- If a slave node fails – it can simply be restarted with no impact on the time critical operations on the master.
- If the master node fails – a single node can quickly take over as there is no need to co-ordinate all nodes.

Network failure

The server can deal with multiple modes of network failure. If the network failure is between the central server and a managed system. More information can be found in the 'Network failure' sections of individual components.

Software failure

If a node fails because of a software induced error (for example out of memory, or a bug in the software stack) then the effect is generally limited to that node. In many cases the failure will not recur as it was the result of a specific action, or specific circumstances. In all setups (but particularly for single node setups) the application server should be configured to restart nodes automatically if they fail, to alert an operator, and to implement a brief delay between restarts.

High availability of the database.

High availability of the database used by the application server is also important, as all data is stored in the database. You should follow the guidelines from your application server vendor for setting up high availability between the application server and the database. Without access to the database many components will wait for it to become available again before continuing, since they cannot write any data.

Database operations are automatically retried if they fail and the failure code indicates that a retrying may fix the error. These retries are done at the transaction level to ensure data integrity.

Database host failure

Should the database host fail, then the database HA solution is responsible for making the database available again. Common solutions include:

- Restarting the database host.
- Fail-over to another host that has access to the same database files
- Active-Active setups like Oracle Real Application Clusters.

In all cases the database HA and the application server are responsible for ensuring that the database is available, and that the data from transactions that have previously been committed is available to SAP CPS. Should this not be the case then data inconsistencies may result.

One important case of database host failure is when the disks (or SAN, etc) that store the database files become unavailable or full. You should ensure that sufficient disk space is available for database data, particularly before significant period end or overnight processing is likely to occur.

Network failure

Transient network failures will be handled by the automatic retry system. This will retry operations using a back-off system that allows brief outages to be recovered from quickly (seconds or less), but will retry less and less often as the length of the network outage increases.

Software failure

The automatic retry system can cope with some software failures, particularly where the failure did not result in a transaction being committed to the database and where retrying the same transaction will result in it being committed successfully. This includes handling errors related to connection pools, some driver errors, and deadlocks in the database caused by row level locking.

Should the database software fail in a more permanent manner then SAP CPS will act as if the database host failed.

High availability of the SAP agent

SAP system failure

The way that the SAP agent has been designed means that it is as resilient as possible against host failure. If the Central Scheduler server goes down, it will not receive further jobs, but the jobs that are already running on the SAP system can finish. The results of these job terminations will be picked up by the SAP agent when the central Scheduler server becomes available and will be applied to the repository (database).

If the SAP system itself goes down, the Central Scheduler server will keep trying to contact the SAP system. There is no timeout or such; it will keep doing this until the SAP system is back up. The Central Scheduler server then examines which jobs were running according to its information, and asks the SAP system what happened to each of these jobs. The SAP agent then in turn determines whether any of these jobs finished. If no information is found, the job outcome is undetermined and the job status is set to *Unknown*. This can then be used to trigger escalation procedures or automatic restarts.

It is recommended to connect the SAP agent to a SAP system via the message server rather than connecting it to a specific ABAP application server as this gives you the following advantages:

- resilience to ABAP application server failures
- automatic load balancing of RFC connections across ABAP application servers

For making the SAP system itself high available you have to consider all its components:

- database
- network
- hardware
- operating system
- software - ABAP application servers, message server, enqueue server, etc

Please refer to SAP documentation and support for more details on making the SAP system high available.

The site http://help.sap.com/ contains information on how to make your specific SAP release high available. It also lists all SPOFs that you have to consider.

Network failure

If the network goes down, but the hosts on both side stay up, the Central Scheduler server is effectively disconnected from the SAP system. It can no longer send new job start requests to the SAP system, nor can job status updates be transmitted to the Central Scheduler server. The server will start reconnecting until the connection is available again. It keeps trying until it succeeds. Once the connection is re-established, operations are picked up where they were left. Some delay will have been introduced, but no status has been lost.

Software failure

Experience so far with the SAP agent shows that it is very reliable, with the highest incidence of customer problems related to configuration issues.

SAP agent fail-over scenarios

The SAP agents run inside the Central Scheduler server, so making the JEE server high available will also make the SAP agents high available.

High availability of platform agents

When examining the availability of platform agents that monitor and control the execution of operating system jobs on the computer systems under the schedulers span of control there are three types of failure that can occur:

- Host failure - what happens if either side of the connection goes down.
- Network failure - what happens if only the network goes down.
- Software failure - crash or kill of the platform agent or just a single component of the platform agent.

Some user setups that guard against the above require *redundancy*: having a fall-back or backup platform agent available. We will show that this can be accomplished in (again) three different manners.

Host failure

The way that the platform agent has been designed means that it is as resilient as possible against host failure. If the Central Scheduler server goes down, it will not receive further jobs, but the jobs that are already running on the remote system can finish. The results of these job terminations are stored on disk until the central Scheduler server is ready to pick this data up and apply it to the repository (database).

If the host itself goes down, the Central Scheduler server will keep trying to contact the platform agent. There is no timeout or such; it will keep doing this until the server that the platform agent runs on is back up and the platform agent has restarted. The Central Scheduler server then examines which jobs were running according to its information, and asks the platform agent what happened to each of these jobs. The platform agent then in turn determines whether any of these jobs finished. If no information is found, the job outcome is undetermined and the job status is set to *Unknown*. This can then be used to trigger escalation procedures or automatic restarts.

Network failure

If the network goes down, but the hosts on both side stay up, the Central Scheduler server is effectively disconnected from the platform agent. It can no longer send new job start requests to the agent host, nor can job termination messages be transmitted to the Central Scheduler server. The server will start reconnecting until the connection is available again. It keeps trying until it succeeds. Once the connection is re-established, operations are picked up where they were left. Some delay will have been introduced, but no status has been lost.

Software failure

The platform agent consists of four different parts, some of which have extended run-times. These parts are:

- A monitor component that has no other task than monitor the network-processor called the platform-agent process.
- The network-processor, which is a long running process that is responsible for communicating with the Central Scheduler server over the network.
- The job-processor, which is started for every job. It starts, monitors and controls a single job.
- The job can call various command-line tools such as jftp, jmail etc.

The command-line tools all serve a single purpose, so there is a low chance of software failures causing long-term effects: every run is stand-alone. If a failure occurs, it only affects that single execution. All tools are designed so that they are easily scriptable, and return reliable error codes.

The job-processor is a little more important, but if it fails this still affects only a single job. The job-processor tries to perform as much as it can before telling the network-processor that it has indeed set the job up as far as possible. It has been designed to do as little as possible after the job has finished. In this way the exposure to software failure that cannot be recovered has been reduced.

The platform-agent process also has a single well-defined and very simple job: monitor the network-processor. As long as the network-processor keeps running it does not do anything. Only when the network-processor stops for any reason does it verify that the network-processor stopped in a controlled manner. If this is not the case this indicates an abnormal failure. In such a case it will sleep for a number of seconds (currently fixed at 10 seconds) and then restart the network-processor. This way it guards against memory errors and other recoverable problems in the network-processor.

The network-processor itself is the most complicated part of the platform agent. It runs multi-threaded, with one task per thread, for an indefinite amount of time. As such there is a higher risk involved with software failure of the network-processor. To keep the actual effect of such a failure low, the network-processor has been designed such that such a failure will not cause a total breakdown of the agent. When the network-processor process aborts completely it is restarted by the platform-agent. When this happens already executing jobs are not affected, as the messages that need to be sent from the job-processor to the Central Scheduler server are stored on disk. The network-processor will simply pick these up when it is restarted.

Experience so far with the platform agent shows that it is very reliable, with the highest incidence of customer problems related to configuration issues.

Platform agent fail-over scenarios

The platform agents normally run on a particular system. If you need redundancy in job execution you are not forced to use a clustered operating system or hardware solution. There are (at least) three scenarios how to handle fail-over between platform agents:

1. Use a clustering solution provided by the hardware or operating system vendor. In such a case you have a single process server and multiple platform agents on the clustered nodes. The clustering software provides virtual IP and file system handover.

2. A single process server, with two platform agents (active and warm) on two different machines. Fail-over from the active to the warm node is handled either by changing the RemoteHostName in the process server (and restarting it) or virtual IP handover.

3. Using two (or more) process servers each contacting a single platform agent. This provides redundancy and enhanced throughput, since it is a warm-warm scenario where both systems can perform useful work if both are functioning properly.

Let us examine these three scenarios in more detail.

Clustering software by 3rd party vendor

The platform agent can run on a system that is part of a hardware cluster. Popular solutions that provide this are IBM HACMP, Sun Cluster, HP ServiceGuard and Microsoft Cluster Service.

Operation of this scenario is as follows: the cluster software determines where the software should run. If this changes, it moves resources to the (winner of the) surviving node(s). Typical resources that are available are virtual IP addresses, hostnames, file systems and software components.

When you intend to use such a solution you must store the agent instance data on a file system that is locally available to the active node in the cluster, and this file system must move with the agent service. The reason that this is important is that the file system is used to persist data about jobs. If the file system is not available, knowledge about jobs that finished on node A will be unavailable once the service moves to node B.

Custom active-warm setup

Instead of a vendor solution you can also prepare a custom setup where a 'warm standby' platform agent is already installed and configured. When problems occur with the active node, it is only a question of either changing the RemoteHostName or changing the IP address of the warm standby host. This does provide a solution for being able to start new jobs, but if you are not able to make the file system where the job status files of the old active node available to the new active node then the job status for those jobs is lost. Whether this is serious depends on your environment; please note that it is not the job outcome itself that is lost, but only the status that the job scheduler has.

Active-active setup

In many circumstances it is possible to set up two or more systems that can each run a particular class of jobs, for instance reporting jobs that connect to a common database, or jobs that perform computations.

The SAP CPS supports such scenarios using the *queue* mechanism. Create a single queue where all jobs of the class of jobs that can run on these systems are sent. Attach multiple process servers, each with a single platform agent, to this queue.

This gives great redundancy characteristics: if a system becomes unusable or unreachable then all remaining jobs will be executed on the remaining process servers. The status of the jobs that was executing on those systems is determined once the connectivity is restored; either the system has crashed in which case the job status becomes *Unknown* or the job has finished successfully and the status was written to (local) disk where it can be retrieved after the failure is repaired.

Note that you can have as many classes of jobs as required, as you can create as many queues as you like, and every process server can work for as many queues as desired. For more information see the Queue mechanism in the Concepts section.

Licensing

As configuring high availability often involves allocating and configuring additional resources, licensing may be affected. Areas where licensing becomes a consideration are:

- When the application is run on multiple server nodes - the license must be valid for all server nodes.
- Where multiple platform agents are being used in an active-active setup - the license must contain a sufficient number of process servers.

If you configure a multi-node system and the license is not valid on all server nodes you will get a warning in the user interface indicating which nodes are not covered by the license. Should this occur, contact your support provider to get an appropriate license.

Configuring Platform Agents on Microsoft Cluster Service

Use

Note

Microsoft Windows process servers are only available when your license includes a non-zero value for the license key *ProcessServerService.OS.limit*. The value of this key determines how many platform agents (across all platforms) you can run.

Using Microsoft Cluster nodes you can create a platform agent that has the same availability characteristics as the Microsoft Cluster. When the node of the platform agent fails or becomes unreachable its service is moved by Microsoft Cluster Service to a backup node.

Cluster Fail Over

- Jobs that are executing when the cluster node moves will be reported as *Unknown*; the system does not know what happened to them.
- A job that finishes immediately before the cluster node moves will get a proper status update to *Completed*, *Error*, or any other final status code (except *Unknown*), when the job-processor still had access to the physical disk at the time when the job finished. If the physical disk resource was already disabled or moved to the winning node in the cluster the job status will be lost and the job will get status *Unknown*.

Platform Agent Network Timeout

A network timeout has been introduced to solve networking issues, when no message has been received in at most twice the MessageInterval, the platform agent will attempt to reconnect.

Cluster Services and Auto Update of the Platform Agent

The Microsoft Windows installer of the Platform Agent creates a service that contains its current version number. When you perform an automatic update, which is pushed out by the server when it detects that the server is newer than the platform agent, a new service with a new name is created. The old service is stopped and deleted.

Platform Agent Network Timeout

A network timeout has been introduced to solve networking issues, when no message has been received in at most twice the {MessageInterval|Cronacle8DocGenProcessParameters], the platform agent will attempt to reconnect.

Installation

- Install the agent in the same directory on the Windows disk drive or another "fixed" path that is identical on all nodes. For example, if you install the agent in `C:\Program Files\Redwood\Agent` then you must install it in this path on every node.
- Create an instance. Give this instance the same name and port number on all nodes.
- Do not register the agent.

After you have installed the agents, make sure that the configuration data is identical, in particular the secret. The standard installation generates different secrets on every node, so you need to override these. This configuration data can be found in the `net` subdirectory relative to where you installed the agent.

It is probably easiest to stop all platform agent instances and copy the `${InstallDir}\net\instance\${Instance}` directory using Windows Explorer or another file copy mechanism.

Prerequisites

To create a service that is managed by the Microsoft Cluster Service you need:

- A Microsoft Windows Server cluster consisting of at least two nodes.
- A shared physical drive that is available on those nodes.

Procedure

1. Install the SAP CPS Platform Agent on all nodes in the cluster, see Creating a Microsoft Windows Process Server for details. Install the product using the guidelines in the *Installation* section above. Make sure you create a process server in the Central Scheduler, such that it uses the cluster IP address and sets DataRootDirectory to a directory on the shared physical drive.

2. Use the Microsoft Cluster Administrator software to add a new resource that is in the same group as the cluster IP address and shared physical disk. Make sure that the platform agent is made dependent on both resources.

3. Start the platform agent (scheduler) resource. The cluster group will show that the Scheduler is off-line. Bring the Platform Agent online. Move the cluster group to a different node to verify that the platform agent service can run on all nodes.

4. Start the process server. Using the SAP CPS UI start the process server. After a few moments the scheduler should go to the *Running* status. If not, check the operator messages. To verify that everything works, submit

and run a job. Move the cluster group to a different node. Run another job. Both jobs should run successfully, but make sure they ran on different nodes.

Values

Here is an example of the data that is filled in

- New Resource

```
Name: Scheduler platform agent <instance>
Description: SAP CPS Job Scheduler platform agent
Resource Type: Generic Service
Group: Suggested: same group as the IP address and physical disk
```

- Possible Owners

```
Add the nodes in the cluster that have the platform agent installed.
```

- Dependencies

```
Add the Cluster IP Address and the physical disk.
```

- Generic Service Parameters

```
Service name: jcs_<version>_<instance>   (MUST be already present on the nodes set in Window 2)
Start parameters: Empty.
```

- Registry Replication is not needed, choose *Finish*.

Organizing Processes

Use

Applications are used to group objects in a hierarchy; they can be nested inside one another.

The following applications are shipped with the product:

- *Redwood* - Parent application for the Redwood modules applications.
- *System* - The System module application.
- *SAP* - The SAP module application.
- *Platform* - The Platform module application.
- *RedwoodScript* - The RedwoodScript module application.
- *JXBP* - The JXBP module application.

Applications have their own section in the navigation bar.

When an application is selected in the navigation bar, the objects in that application are shown in the main pane ordered by the type of object. This allows you to quickly and easily locate an object or a group of objects.

Applications can contain other applications. This means that you cannot delete an application which has child applications or contains objects. You must first remove the contained objects from the application. This can be done by deleting contained objects, moving them to another application (by changing the value in each object's *Application* field), or setting their application to null.

Additionally, applications can be exported to a CAR archive file, using job definition System_ExportApplication or manually. These archives can then be imported with the System_Import_Archive job definition into other Repositories. This allows for easy migration from a DEV or QA environment into a PROD environment.

Actions

Applications support the following actions:

Action	**Description**
Export tree	Export the application tree, including all child objects of the selected application
Monitor Related Jobs	Go to the job monitor, enable a temporary filter showing all jobs related to the application
Edit	Edit the application
Edit Security	Edit Security of the application
Delete	Delete the application
Duplicate	Duplicate the application
Expand All	Expand all applications
New	Create an application

Deleting Applications

You can only delete libraries when no other objects relate to them. For example, if there are job definitions that use the application, the application cannot be deleted until all job definitions that use it have been modified. You can see all job definitions that relate to the application in *Related Objects* in the lower detail pane and on the show page.

The table in related objects contains three columns:

- Type - the type of object with a link to it
- Related Object - the name of the object with a link to it
- Used As - objects can sometimes be used in different roles

Security

Privilege	**Description**
Application.Create	Create applications
Application.View	view applications
Application.CreateChildren	Create child-applications
Application.Edit	Edit applications

Application.Delete	Delete applications

You can grant privileges on two levels, *Access* and *Admin;* a privilege granted on *Admin* level allows the grantee to grant the privilege to other users. These privileges can be granted *system-wide*, per *partition* or *isolation group*.

If you have the security module, which requires the *Module.Security* license key, you have an additional *Security* tab on the application. It allows you to specify which users can access, edit, create children of, and delete the application.

- Granting and Revoking System Privileges
- Granting or Revoking Object Privileges

See Also

- Creating Applications

Creating Applications

Use

Applications are used to group objects used to group other objects in a hierarchy; they can be nested inside one another.

Procedure

To create an application:

1. Choose *Applications*.
2. From the context menu of the left-hand side bar, choose *Create* .
3. Fill in the necessary information, refer to the values below.

Values

- *Name* - The name of the application.
- *Parent Application* - The name of the application that contains this application.
- *Description* - The description of the application.
- *Comment* - A comment about the application.

Example

1. Choose *Applications* in the Navigation bar.
2. From the context menu, choose *Create* , then fill in the values below.
3. Choose *Save & Close* to save the new application and close the edit window.

```
Name: EMEA_DataWareHouse
Parent Application: [none]
Description: Data warehouse for EMEA
```

```
Comment: This application is used to group all objects related to the EMEA Data Warehouse.
```

Managing Applications with RedwoodScript

Use

You can use RedwoodScript to manage applications. ApplicationObjects are objects that can have an application and the application can be retrieved by the getParentApplication() method. This differs from most other properties, because of the *Parent* part of the method name. This has been done, so that the method name matches the name of the method you use on application objects.

Example

You want to move all objects from the App1 application to the App2 application.

```
{
Application oldApp = jcsSession.getApplicationByName("App1");
Application newApp = jcsSession.getApplicationByName("App2");
for (Iterator it = oldApp.getChildApplications() ; it.hasNext();)
    {
      ApplicationObject o = (ApplicationObject) it.next();
      o.setParentApplication(newApp);
      jcsSession.persist();
    }
}
```

You want to force users to specify an application when ApplicationObjects (except Applications) are created. (Note that ApplicationObjects are objects which can have a parent application). This can be used as a trigger.

```
{
  Application parentApp = jcsSession.getApplicationByName("Masalan");
  Application app = jcsSession.getApplicationByName(parentApp, "Finance");
  for (final Iterator it = jcsImportRuleSet.getObjects().iterator(); it.hasNext();)
  {
    Object o = it.next();
    if (o instanceof ApplicationObject && !( o instanceof Application))
    {
      ((ApplicationObject) o).setParentApplication(app);
    }
  }
}
```

Importing and Exporting SAP CPS Objects

Use

Most objects can be exported as CAR files from within the user interface, they can then be imported with the job definition *System_Import_Archive* or from the command line. To customize your imports and exports, you can specify rule sets. Import and export rule sets allow you to organize the flow of processes between Development, Quality Assurance and Production systems.

Actions

Import rule sets support the following actions:

Action	Description
Export	Export the import rule set into a CAR file
Submit Import Job	Submits an import job with the current import rule set
Edit	Edit the import rule set
Edit Security	Edit the security of the import rule set
Delete	Delete the import rule set
Duplicate	Make a copy of the import rule set to create a similar one
Expand All	Expand all import rule sets in the current filter
New	Create a new import rule set
Filter > New Filter	Create a new import rule set filter
Filter > Edit Filter	Edit current import rule set filter
Filter > Delete	Delete current import rule set filter

Export rule sets support the following actions:

Action	Description
Export	Export the export rule set into a CAR file
Submit Export Job	Submits an export job with the current export rule set
Edit	Edit the export rule set
Edit Security	Edit the security of the export rule set
Delete	Delete the export rule set
Duplicate	Make a copy of the export rule set to create a similar one
Expand All	Expand all export rule sets in the current filter
New	Create a new export rule set
Filter > New Filter	Create a new export rule set filter
Filter > Edit Filter	Edit current export rule set filter
Filter > Delete	Delete current export rule set filter

Imports support the following actions:

Action	Description
Edit	Edit the import
Delete	Delete the import
Submit Import Job	Submits an import job
Expand All	Expand all imports in the current filter
Filter > New Filter	Create a new import filter
Filter > Edit Filter	Edit current import filter
Filter > Delete	Delete current import filter

Exports support the following actions:

Action	Description
Edit	Edit the export

Delete	Delete the export
Submit Export Job	Submits an export job
Expand All	Expand all exports in the current filter
Filter > New Filter	Create a new export filter
Filter > Edit Filter	Edit current export filter
Filter > Delete	Delete current export filter

Finding Import and Export Rule Sets

You can search for import and export rule sets using filters and the *Search Import Rule Sets* and *Search Export Rule Sets* box, respectively, known as the *intelliSearch* boxes, located under your username on the top right-hand side of the user interface. Filters allow you to specify a list of objects with static criteria. IntelliSearch allows you to specify complex queries in a simple way using prefixes. Prefixes are used to specify which property you are searching in and have short and long syntaxes. For example, if you want to display all import rule sets with the term *dev* in the comment, you would use the search criteria as follows:

```
c:dev
```

You can search more than one property, as follows:

```
c:dev n:ORA
```

Note

No spaces should be entered before or after the colon (*:*).

See the Advanced Object Search for more information.

The following table illustrates the available prefixes for import and export rule sets:

Prefixes	**Description**
n, name	searches the name property
c, com, comment	searches the comment property
d, desc, description	searches the description property
a, app, application	searches the application property

Import Rule Sets

Import rules define which objects get imported from CAR archives and the changes required by the new system, which allow one or more objects to be transported from one system into another seamlessly. Import rule sets can hold one or more import rules. When you create an import, you may specify an import rule set according to which Parameters, like `INSTANCE`, `CLIENT,` or `USERNAME` for SAP job definitions, can be adjusted to the new system with overrides.

All objects are replaced when you import them except for tables. When you import a *System_Variables* table, new rows get the *Variable Prefix* prefixed, modified rules are not changed.

Note

Imports, import rules and import rule sets require the Import/Export module, which requires a specific license.

Actions on import rule sets allow you to specify RedwoodScript code to be executed against the imported objects before they are written to the database. The code can be specified on the *Actions* tab and make use of the classes that you have stored in a library.

Note

You should not persist the session in import rule set actions, as this is done for you at a later stage.

Export Rule Sets

Export rules define which objects get exported into CAR archives. You can match one or more objects to be transported from one system into another seamlessly. Export rule sets can hold one or more export rules.

There is no need to exclude built-in objects, as they cannot be exported and will be discarded.

Note

Built-in objects cannot be exported, as such, any built-in object that matches a pattern will be discarded from the export.

Note

Exports, export rules and export rule sets require the Import/Export module, which requires a specific license.

Importing Objects from Remote CAR Files

A CAR file is located on a remote system, you would like to import it into SAP CPS. You have multiple choices, for example, you can:

- run an import and use the file browser to upload the CAR file
- copy the file over to the target central SAP CPS server and submit *System_Import_Archive*, specifying the local path to the file, for example `c:\files\Masalan.car`
- copy the file to a web server where it can be downloaded and submit *System_Import_Archive*, specifying the HTTP URL to the file, for example `http://myserver.masalan.com/Masalan.car`
- execute the car file with `java` (on the command line) and specify the URL to the central server (see section below)

Importing CAR Files from the command line

The uploader class is included in the CAR file, so you can run upload CAR files from the command line interface.

```
java -jar <CAR> [-server [http[s]]://]host:port -username <username> -password <password> <option> <arguments

if -server is omitted, you will be prompted for the data

Valid <options>
 -gui (present a GUI for entering server/username/password)
 -txt (command line prompt for server/username/password)

If the -gui/-txt option is omitted the input method will depend on the operating system and JDK version.

Valid <arguments>
 -self - self upload this file (default)
 -ruleset <ruleset name> - use an import ruleset (optional)
```

Example

```
java -jar processServer.car -server prd.masalan.com:50000 -username jdoe -password mysecret
```

Security

Privilege	Description
ImportRuleSet.Create	Create import rule sets
ImportRuleSet.Delete	Delete import rule sets
ImportRuleSet.Edit	Edit import rule sets
ImportRuleSet.View	Access import rule sets
ExportRuleSet.Create	Create export rule sets
ExportRuleSet.Delete	Delete export rule sets
ExportRuleSet.Edit	Edit export rule sets
ExportRuleSet.View	Access export rule sets

You can grant privileges on two levels, *Access* and *Admin;* a privilege granted on *Admin* level allows the grantee to grant the privilege to other users. These privileges can be granted *system-wide*, per *partition* or *isolation group*.

If you have the security module, which requires the *Module.Security* license key, you have an additional *Security* tab on the import and export rule set. It allows you to specify which users can access, edit, and delete the import or export rule set.

- Granting and Revoking System Privileges
- Granting or Revoking Object Privileges

Procedure

Creating Import Rule Sets

1. Navigate to *Promotion > Import Rule Sets*
2. Choose *New* from the context menu.
3. Fill in the details and choose the *Import Rules* Tab
4. Choose *New* to create a new import rule and fill in the details.
5. Choose *Save & Close*.

Using Import Rule Sets

1. Navigate to *Promotion > Import*

2. Choose *Import* from the context menu.

3. Fill in the details and specify an import rule set to narrow down the imported objects.

4. Choose *Save & Close*.

Creating Export Rule Sets

1. Navigate to *Promotion* > *Export Rule Sets*

2. Choose *Create* from the context menu.

3. Fill in the details and choose the *Export Rules* Tab

4. Choose *New* to create a new import rule and fill in the details.

5. Choose *Save & Close*.

Using Export Rule Sets

1. Navigate to *Promotion* > *Export*

2. Choose *Export* from the context menu.

3. Fill in the details and specify an export rule set.

4. Choose *Save & Close*.

Example

Using import rules to change objects during import

A CAR archive containing multiple job definitions are to be imported, these job definitions all have a parameter named `SAP_SYSTEMS` that needs to get a new default value. The default value in the car file is *Q2A* and should be replaced with *PR2* to match the new system. The parameter `CLIENT` also needs to change from *200* to *800*.

1. Navigate to *Promotion* > *Import Rule Sets*

2. Choose *Create* from the context menu.

3. Fill in a name and choose the *Import Rules* tab.

4. Choose *New* to create a new import rule and fill the details below into the *Name* and *Object Type* fields.

5. Choose *New* next to *Parameter* for the *Matching* field and fill in the details below, do the same for *Replaces*.a

6. Perform steps 3 to 5 for the *CLIENT* parameter.

7. Choose *Save & Close*.

Values

SAP_SYSTEMS parameter:

Field	**Value**
Name	Q2A_TO_PR2

Object Type	Job Definition
Matching - Name	SAP_SYSTEMS
Matching - Default Expression	Q2A
Replaces - Name	SAP_SYSTEMS
Replaces - Default Expression	PR2

CLIENT parameter:

Field	**Value**
Name	200_TO_800
Object Type	Job Definition
Matching - Name	CLIENT
Matching - Default Expression	200
Replaces - Name	CLIENT
Replaces - Default Expression	800

Using RedwoodScript code to change objects on import

1. Navigate to *Applications*.
2. Choose *Create* from the context menu and fill `Masalan` into the *Name* field.
3. Choose *Save & Close*.
4. Navigate to *Promotion > Import Rule Sets*
5. Choose *Create* from the context menu.
6. Fill in a name and choose the *Import Action* tab.
7. Choose *New* and fill the code below into the *Source* field.
8. Choose *Save & Close*.

Sample code to set the application or parent application to the `Masalan` application

```
{
  Application app =
jcsSession.getApplicationByName("Masalan");
  for (final Iterator it = jcsImportRuleSet.getObjects().iterator();
it.hasNext();)
  {
    Object o = it.next();
    if (o instanceof ApplicationObject)
    {
      ((ApplicationObject) o).setParentApplication(app);
    }
  }
}
```

Managing Job Output Formats

Use

Jobs produce different kinds of output, the file type can range from plain text files to PDF or ZIP files. To allow the browser to use the correct application to open the output file, a MIME type needs to be specified. Formats can be attached to job definitions and job chains to specify the output format and MIME type.

A format describes file content types, such as text or Postscript. A format can be assigned to a job definition's output or log format, thus defining the extension and MIME (*M*ultimedia *I*nternet *M*ail *E*xtensions) type to use for the job files. The application used to view a job file depends on your browser, and your browser settings.

Formats are part of the Definitions section in the navigation bar. The following standard formats are provided with SAP CPS:

Format	**Explanation**	**Extension**	**MIME type**
CSV	Comma-separated Values files	csv	text/csv
CronacleArchive	Cronacle Archive files	car	application/x-vnd.redwood.car
HTMLText	HyperText Markup Language files	html	text/html
JavaArchive	Java archive files	jar	application/java-archive
PlainText	Plain Text or text files with no formatting	txt	text/plain
PortableDocumentFormat	Portable Document Format files	pdf	application/pdf
TextLog	Text Log files	log	text/plain
XMLText	eXtensible Markup Language files	xml	text/xml

Feel free to add any formats that your jobs generate.

Actions

Formats support the following actions:

Action	**Description**
Export	Export the format into a CAR file
Edit	Edit the format
Edit Security	Edit the security of the format
Delete	Delete the format
Duplicate	Make a copy of the format to create a similar one
Expand All	Expand all formats in the current filter
New	Create a new format
Filter > New Filter	Create a new format filter
Filter > Edit Filter	Edit current format filter
Filter > Delete	Delete current format filter

Mapping of files to formats

The system's process server agents will try to assign a format to every job file created by the jobs. For some files the system knows exactly what the content is as it has generated them itself, for instance the agent log file. For others, it maps the extension of the job's filename to the list of possible formats. If that fails it uses the *Default Output Format* and *Default Log Format* attributes of the Job Definition. If those are not set it uses PlainText or TextLog. The difference between PlainText and TextLog files is small at the moment, but a future version of SAP CPS will allow smart browsing through TextLog files.

System jobs

For System jobs, the job output is in PlainText format. The agent's error messages are in TextLog format.

SAP jobs

For SAP jobs the job log, the application logs and the job statistics log is in TextLog format. The default format of the spool files is PlainText. If the SAP system supports new XBP functionality for retrieving spool as PDF (see SAP Note 1280850), then the spool output of type OTF (SAP Script) is automatically converted into PDF, other types of spool will be retrieved as PlainText. If the *Default Output Format* of the SAP job definition is set to *PortableDocumentFormat*, then all spool files will be in PDF.

Operating system jobs

For Operating System jobs run by the Platform Agent, the job agent log (stdlog.log) is in TextLog format. The job's standard error stream (stderr.log) is in the job definition's Default Log Format, or if that is not set in TextLog format. The job's standard output stream (stdout.log) is in the job definition's Default Out Format, or else in Text format. Other files are mapped according to their extension. If this fails, the format is set to the job definition's Default Out Format, or else to Text format.

Finding Formats

You can search for formats using filters and the *Search Formats* box on the Formats tab. This box is known as the *intelliSearch* box and located under your username on the top right-hand side of the user interface. Filters allow you to specify a list of objects with static criteria. IntelliSearch allows you to specify complex queries in a simple way using prefixes. Prefixes are used to specify which property you are searching in and have short and long syntaxes. For example, if you want to display all formats with the term *W3C* in the comment, you would use the search criteria as follows:

```
c:W3C
```

You can search more than one property, as follows:

```
c:W3C n:XML
```

Note

No spaces should be entered before or after the colon (*:*).

See the Advanced Object Search for more information.

The following table illustrates the available prefixes for formats:

Prefixes	Description
n, name	searches the name property
c, com, comment	searches the comment property
d, desc, description	searches the description property
a, app, application	searches the application property

Deleting Formats

You can only delete formats when no other objects relate to them. For example, if there are still jobs or job definitions that reference the format, the format cannot be deleted until all references have been deleted. You can see all references to the format in *Related Objects* in the lower detail pane and on the show page.

The table in related objects contains three columns:

- Type - the type of object, like *Job Definition* for a job definition, with a link to it
- Related Object - the name of the object with a link to it
- Used As - objects can sometimes be used as different object types

For example, formats that are used in job job definitions, will have the call-reference in *Related Objects*. The type will be *Job Definition*, for example and the *Related Object* will be the name of the job definition.

Note

When a job definition is displayed with a timestamp in the name, this indicates that there still is a job that refers to the format. You might have changed the job definition, but as long as that job is in the system you will not be able to delete the format.

Security

Privilege	Description
Format.Create	Create formats
Format.Delete	Delete formats
Format.Edit	Edit formats
Format.View	Access formats

You can grant privileges on two levels, *Access* and *Admin;* a privilege granted on *Admin* level allows the grantee to grant the privilege to other users. These privileges can be granted *system-wide*, per *partition* or *isolation group*.

If you have the security module, which requires the *Module.Security* license key, you have an additional *Security* tab on the format. It allows you to specify which users can access, edit, and delete the format.

- Granting and Revoking System Privileges
- Granting or Revoking Object Privileges

Procedure

1. Choose *Definitions* > *Formats*.
2. Choose *Refresh* to see all existing formats.

3. Choose *New* to create a format.
4. Fill in the details, refer to the section below.
5. Choose *Save and Close* to save the format.

Format

- *Name* - The name of the format.
- *Application* - The name of the application used to group this object.
- *Description* - The description of the format.
- *Comment* - A comment about the format.
- *Extension* - The extension of files that use this format.
- *MIME Type* - The MIME type to use when sending this file to a web browser.

Example

1. Choose *Definitions* > *Formats* in the Navigation bar.
2. Choose *New* and fill in the values below.
3. Choose *Save & Close* to save the new format and close the edit window.

```
Name: ZIP
Description: Compressed ZIP File
Comment: Compressed ZIP file containing other files
Extension: zip
Mime Type: application/zip
```

See Also

- Working with Job definitions

Regular Activities for SAP CPS

Use

All housekeeping has been scheduled to run in intervals of 15 minutes by default, which should be sufficient for most systems. When your system is very busy and one or more of the following jobs runs for too long or uses too many system resources at any given point in time, you might want to consider scheduling these jobs less frequently or at specific times.

- System_UpdateJobStatistics - updates the runtime statistics of jobs
- System_ProcessKeepClauses - calculates the keep clauses for jobs, to determine if a job has to be deleted or not
- System_DeleteJobFiles - deletes the output and log files of deleted jobs

Please note that changing the periodicity of these jobs can have the following consequences:

- statistics data will tend to be outdated the more you widen the interval
- keep clauses will be honored late the more you widen the interval which can affect disk space and overall performance
- the total amount of disk space used by jobs for output and logs at peak times will increase, since the output is deleted later files will pile-up

The Following submit frames are associated with the above job definitions and should be used to influence their periodicity:

- System_UpdateJobStatisticsSubmitFrame
- System_ProcessKeepClausesSubmitFrame
- System_DeleteJobFilesSubmitFrame

See the Tuning SAP CPS and Submit Frames sections for instructions on changing the submit frames.

Automatically Deleting Jobs

You control the size of the job history (number of jobs in the repository) with the keep clauses which can be specified on the job definition. By default, all jobs are kept, it is however strongly recommended to specify a keep clause for every job that is to be scheduled regularly.

Deleting Jobs

There are two basic ways of deleting jobs in the user interface:

- You can select one or more in the job monitor (the same way you would on your platform, for example, hold down the *Shift* key as you choose on Microsoft Windows) and use the context-menu.
- You can delete all completed jobs of a job definition using the context-menu of the job definition.

When you are in an emergency situation and you need to delete a great number of old jobs to free-up disk space or increase database performance, you can use the job definition System_RemoveOldJobs. Please do not schedule the job definition regularly, use keep clauses instead.

Cleaning Ghost Records up MS SQL Server Databases

Customers using a great number of jobs with many parameters and short keep clauses can experience the build-up of ghost records in the indexes on `JCS_JOBPARAMETER0`. This can slow-down the system and use up a tremendous amount of space. If you cannot reduce the number of job parameters, you will have to rebuild the indexes regularly on that table; the frequency depends on the volume of job parameters.

You schedule a rebuild of the indexes at regular intervals, using the following commands, for example:

```
ALTER INDEX PK__JCS_JOBP__D9CE5306737017C0 ON JCS_JOBPARAMETER0 REBUILD
ALTER INDEX JCS_JOBPARAMETER00 ON JCS_JOBPARAMETER0 REBUILD
ALTER INDEX JCS_JOBPARAMETER01 ON JCS_JOBPARAMETER0 REBUILD
```

Note

Please check the names of the indexes in your database.

Backup and Recovery of SAP CPS

Use

This document explains where SAP CPS stores data and how to handle backup, restore and disaster recovery for that data. It covers which sorts of data are stored, where, and appropriate backup & restore strategies for each data set. It covers backup for SAP CPS itself, not for scheduled applications. It is intended for operational use, not for archiving.

Architecture

SAP CPS stores data locally to meet two main operational requirements:

Writing to local files is independent of the network infrastructure.

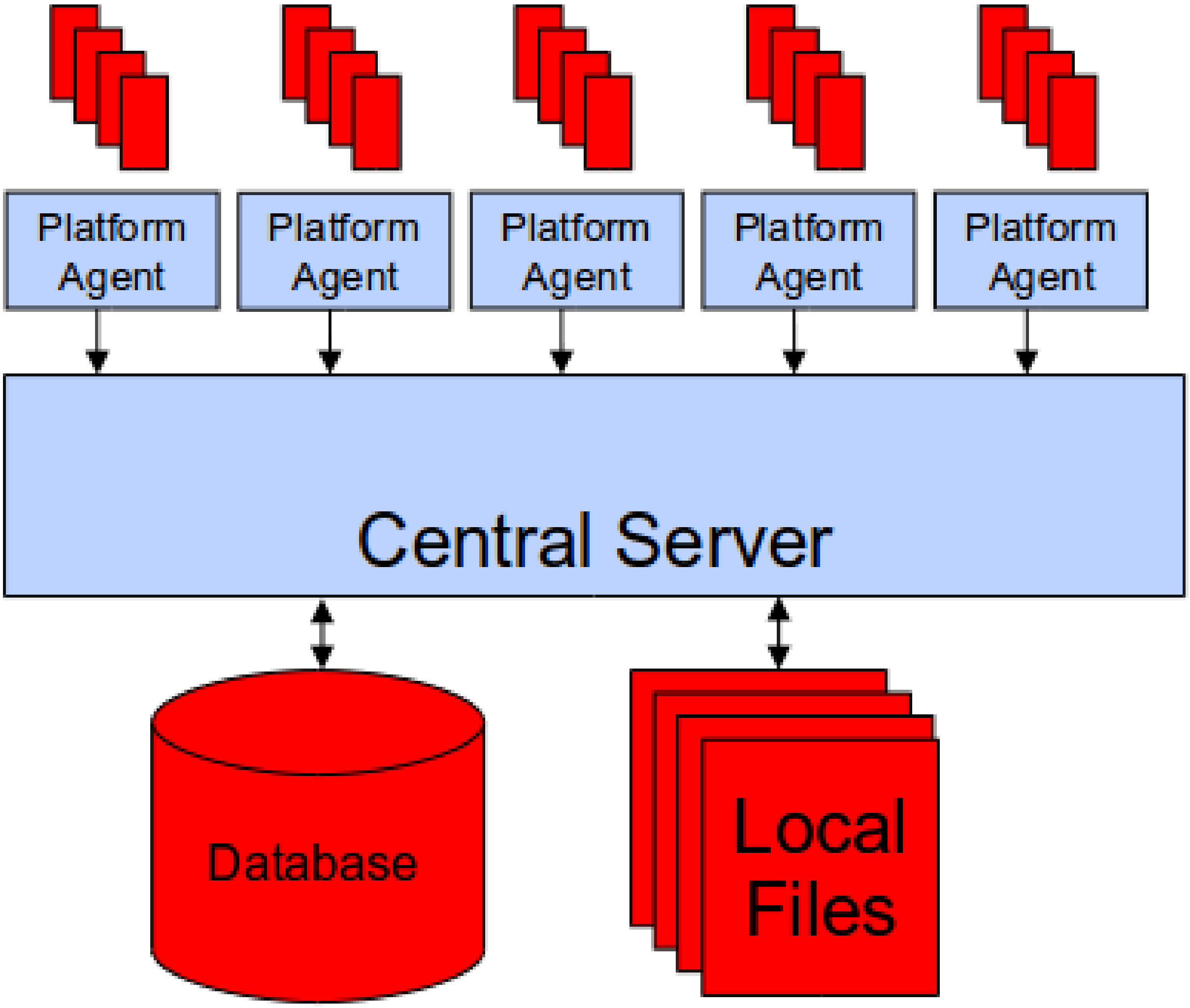
Local writes are generally faster than network writes.

There are five types of data stored:

- Software – the code and data that make up the product.
- Local traces – diagnostic information.
- Local persistent data – local copies of data to be sent to the server.
- Job Logs and Output – the output of jobs run on a certain machine.
- Database data – the central store used for server data.

Data Locations

The diagram below shows where the data (Red) and software (Blue) are stored:



Backup

The backup strategy and schedule differ per location and type of data:

Type	Location	Strategy	Frequency
Software	All	File	Before/after upgrades
Traces	All	File	Daily
Local persistent	All	File	Daily
Job output & logs	All	File	Daily
Database Data	Database	Database	Daily

Upgrades

You should back up all data before and after doing an upgrade or patch. The backup before the upgrade can be used to restore the system to a working state should the upgrade or patch fail or have undesired consequences. The backup after the upgrade ensures that you do not have to perform the patch or upgrade again in the case of a restore being required.

File Backup

This document assumes that your file backup software can backup programs while they are running. Such software has two advantages: a) normal operations can continue, b) you can schedule the backups with SAP CPS.

If you cannot do this, then you will need to shut down the entity being backed up. For the central server this is the web application server, for platform agents you need only shut down the platform agent itself (the process server will go into *Connecting* status until the agent returns).

When backing up the software you should back up:

- Server – the entire web application server installation (including other applications)
- Platform agent – the installation directory that contains the 0_nn_mm, net, etc and var directories. For a software only backup you do not need to backup the var directory (this contains data).

When backing up the local persistent data and job output & logs you should backup the directory pointed to by the process server parameter *DataRootDirectory*. For the local files on the central server, some files are stored in the (optional) 'DataRootDirectory' of the System process server, the default value of this parameter will vary, consult the guide for your application server for more information.

Backups of job output and logs may contain partial files for jobs that are running at the time. Your backup software can either be configured not to backup open files, or left to backup these files. Should you do the latter then only the partial output will be restored.

Database Backup

See the Database Backup topic for more information.

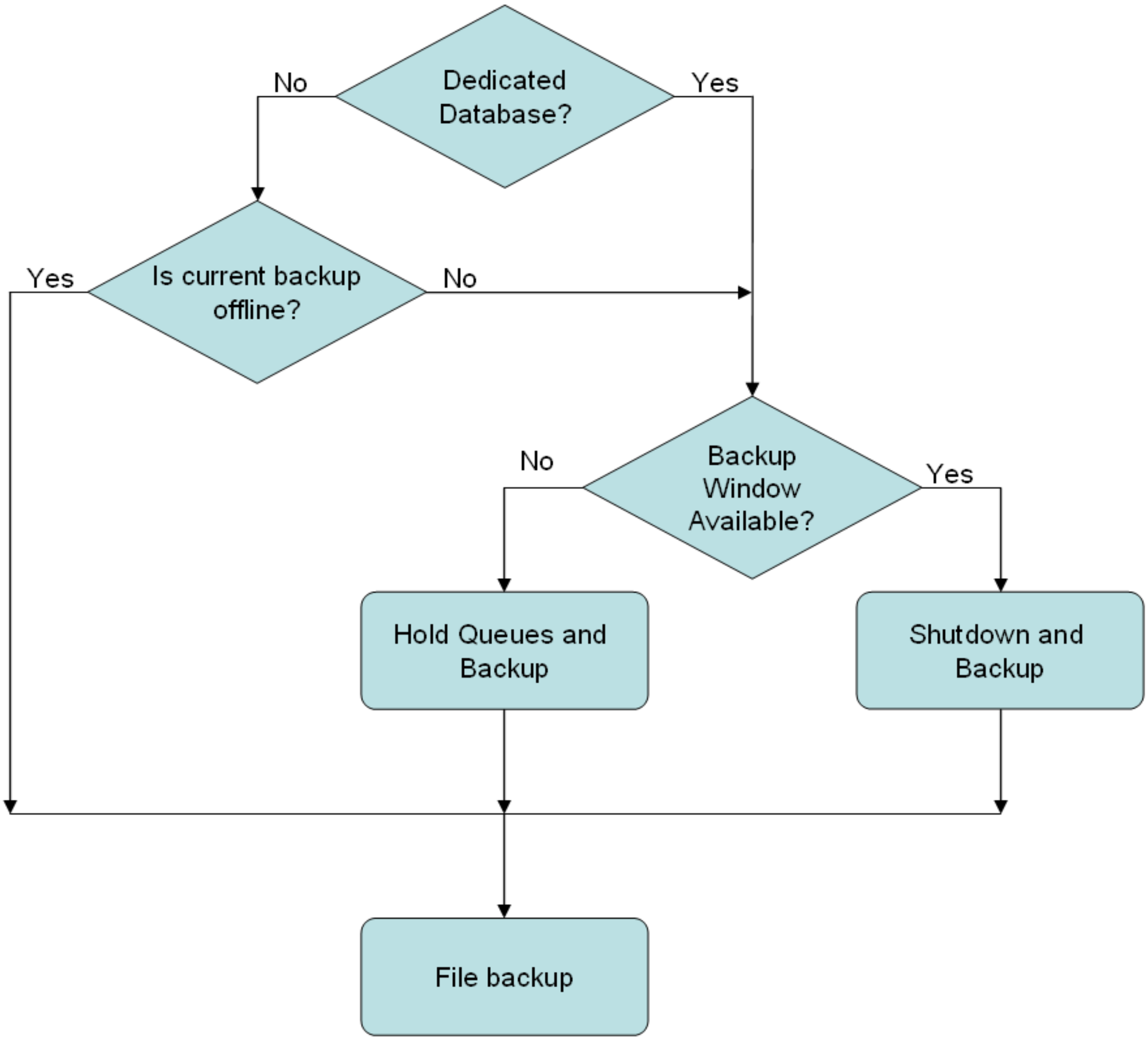
See Also

- Database Backup topic
- Restore Data

Database Backup

Use

We recommend an online database backup is performed daily, ideally when no jobs are running as this will simplify restore procedures. Should no suitable window be available, then a backup with the minimal number of running jobs will keep additional restoration activities to a minimum.



Database backup – Shutdown and Backup

Ensuring that no jobs are running simplifies both the backup and restore procedures. The following procedure can be used to ensure that no jobs are running:

1. Hold all queues.
2. Wait for all jobs to complete.
3. Shut down all process servers (except the System process server).
4. Perform the backup.
5. Start all process servers
6. Release all queues.

Database backup – Hold Queues and Backup

If it is not possible to perform a backup with no jobs running, then holding the queues will prevent new jobs from starting, and ensure that the system will not automatically start new jobs when it is restored from a backup. The following process can be used:

1. Hold all queues.
2. Perform the backup.
3. Release all queues.

Database backup – Other applications

If other applications (including different instances or older versions of SAP CPS) are present in the database, then it may be desirable to be able to back up only the data for a specific instance of SAP CPS, rather than backing up the entire database, particularly if the other applications store the majority of the data.

SAP CPS stores only tables and indexes in the database, it does not use any proprietary objects (for example public synonyms, views, stored procedures, or triggers). All tables and views have the prefix *JCS_*. If multiple users or schema have `JCS_` tables then you will need to know which database the data are stored in and which username is used to access them.

You need to back up all tables in that database that are owned by the username (or schema for DB2).

Database backup – Multiple instances

If you have multiple instances in the same database, then they need to be installed and accessed as different usernames (schema for DB2). In this situation you can either back up all instances using a database backup, or back them up individually. You will need to ensure that you are working with the correct instance/username during all backup, restore and disaster recovery activities.

If you are using Oracle and have one of the Oracle based versions of SAP CPS some additional considerations apply. You can check if you have such an older version by looking for a public synonym for a package called *jcs*. If you have one of these older versions, then you cannot install the SAP CPS under the same user (generally SYSJCS)

Restore Data from Backup

Use

The main factors to consider when restoring SAP CPS from a backup are:

- Consistency of the restoration across the entire landscape.
- Reconciliation of jobs that are running or were scheduled between the backup and restore times.

Consistency

As data is stored in many locations in the system landscape, consistency of that data can be an issue. For instance, job information like Status is stored in the repository, while job output is stored on disk. If the backup of the database is older than the disk then there will be job output files on disk that are not linked to the job, while if the database is newer than the disk then the job may be *Completed* with links to job files that do not exist.

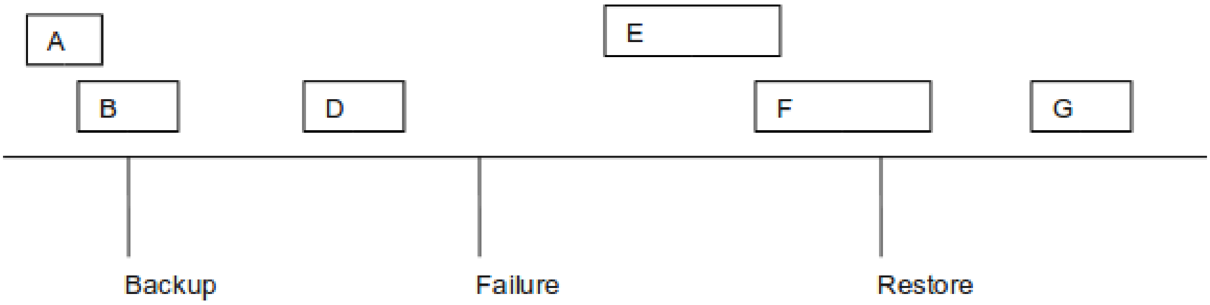
Consistency problems can be reduced by:

- Frequent backups.
- Performing backups when no jobs are running.

Reconciliation

Even if you have a consistent restore, it is possible that jobs were scheduled to run (and in fact ran) between the time of the backup, the time of the failure and the time that the restore occurred. This means that a database restore may not contain correct for this period.

The timeline below illustrates the possibilities:



The different jobs start and finish (or are scheduled to) at different times relative to the backup, failure and restore, and hence require reconciliation strategies:

Type	Reconciliation
A	None required – captured completely by backup. Status and output will both be correct.
B	Job will appear as Running and change status when the process server is started. May resolve automatically for: SAP jobs that still exist on the managed system. Platform agent jobs that are still running. Otherwise the status will be set to unknown.
C	Will appear as *Scheduled* even though it started and finished. Can be left to re-run automatically if there are no side-effects. If it is critical that the job is only run once, manual resolution is required. May only have partial job files present.
D	Ran and had its status updated, but information was lost because it was not part of the backup. Resolution for status is the same as C, job files will be complete.
E & F	These jobs will be in status *Scheduled*, and not have been started, they will start when the system is restored.
G	These jobs are unaffected.

In all cases, if the job started before the failure and the system where the job was running was not affected by the failure, then job output will exist and be correct.

Reconciliation problems can be reduced by:

- Performing backups when no jobs are running – this means that there will be no jobs of type B.
- Holding queues while the backup is performed, or setting process servers not to start on startup – this will prevent jobs of type E and F from starting automatically when the restore completes.

Frequent backups – this will reduce the number of jobs of type C, D, E and F and hence reduce the amount of reconciliation work.

Chapter 8. Automating Processes Across the System Landscape

For process automation, there are various repository objects that can be viewed and edited in the Definitions tab. Once you have setup your scheduling landscape with connections to the external systems you would like to use, you can create processes with the following repository objects:

- Job Definitions - A job definition is a repository object that defines a process to be executed by the process server. A job definition consists of command text to be executed and several runtime and scheduling controls.
- Job Chains - A job chain is built from a set of calls to job definitions (named job chain jobs) grouped in one or more steps.
- Event Definitions - An event definition is a repository object that defines event objects. Event objects are used by other job definitions and job chains to signal that a job must start or has completed.
- Formats - Jobs and job chain jobs produce output files in various formats, to enable client systems to use the correct program to open output files, the correct format must be created and defined in the job definition.
- Resources - A resource is an object that defines which resources (programs, libraries) are needed by a process server to execute a job. Process servers provide resources.
- Locks - A lock is used to prevent two jobs from running simultaneously. When jobs share a lock, the first job to start will hold the lock, preventing other job definitions with this lock from starting until it completes.
- Reports - Generate reports based on queries in HTML or Comma Separated Values (csv) format

1. Creating and Editing Job Definitions
2. Creating and Editing Job Chains
3. Submitting Jobs and Job Chains
4. Scheduling with Times and Dates
5. Importing Objects from SAP Systems
6. Triggering Custom Code
7. Scripting in the Shell

Creating and Editing Job Definitions

Use

A job definition is a Repository object that defines a process to be executed by the process server. Every job that is submitted is the request of the execution of a single job definition. SAP CPS ships with a number of job definition types, which define the programming language and target platform of the commands to be executed.

A job can raise one or more events when the job gets a specific job status or return code. The events to be raised can be specified in the job definition.

Job definitions can be made dependent upon specific events. These jobs cannot be executed before the events upon which they are dependent are raised. They will remain in *Event Wait* status until all dependent events are raised.

Parameters enable job definitions to define generic command texts that behave differently, according to the values of the job definition parameters. For these parameters, validation rules can be added by defining simple constraints on the parameters.

Restart behavior is used to break or continue the submit frame when something unexpected happens. When you break the submit frame, the job definition or job chain does not get submitted again according to the submit frame. By default, a *Completed* job definition or job chain will continue to get submitted according to the submit frame.

Special care needs to be taken when you edit a job definition which is currently being (re)scheduled; the new job definition will take effect on all jobs that have not reached the *Dispatched* job status at modification time. Please see the Editing Job Definitions Safely section.

Actions

Job definitions have the following actions:

Action	Description
Submit	Submit the job definition
Cancel non-final state jobs	Cancel jobs that have not reached a final state, see the Job States section for more information on job states.
Delete final state jobs	Delete jobs that have reached a final state, see the Job States section for more information on job states.
Monitor related jobs	Jump to the job monitor with a temporary job filter displaying jobs for the job definition
Reset Statistics	Reset the runtime statistics for the job definition
Export	Export the job definition, only custom job definition definitions can be exported
Edit	Edit the job definition, only custom job definitions can be edited
Delete	Delete the job definition, only custom job definitions can be deleted
Duplicate	Duplicate the job definition, only custom and some built-in jobs can be duplicated
Expand all	Mainly used for jobs chains, allows you to expand the entire tree of the job chain
New	Create a new job definition
New (from Template)	Create a new job definition from a template
Filter > New Filter	Create a new job definition filter
Filter > Edit Filter	Edit an existing job definition filter
Filter > Delete Filter	Delete a job definition filter
Help	Display the help topic(s) covering job definitions

Compulsory Fields

Some fields need to be filled, like the *Job Definition Name* field, which is the only field that does not have a default and that needs to be filled for every job definition type. Other job definition types additionally require other fields be filled. For example, RedwoodScript job definitions need to have the *Source* field filled with at least the following: `{}`. Furthermore, the source is not editable with job definitions of certain types, like the *SAPR3* job definition type.

The keep clause, which by default is set to keep all jobs of the job definition, is also compulsory and needs to be changed if you want the system to automatically delete old jobs.

Note

The source cannot be edited for all job definition types, other job definition types require a specific license key. See the Editing the Job Definition Source section for more information.

Finding Job Definitions

You can search for job definitions using filters and the *Search Job Definitions* box on the Job Definitions tab. This box is known as the *intelliSearch* box and located under your username on the top right-hand side of the user interface. Filters allow you to specify a list of objects with static criteria. IntelliSearch allows you to specify complex queries in a simple way using prefixes. Prefixes are used to specify which property you are searching in and have short and long syntaxes. For example, if you want to display all job definitions with the term *import* in the comment, you would use the search criteria as follows:

```
c:import
```

You can search more than one property, as follows:

```
c:import n:System
```

Note

No spaces should be entered before or after the colon (*:*).

See the Advanced Object Search for more information.

The following table illustrates the available prefixes for job definitions:

Prefixes	**Description**
n, name	searches the name property
c, com, comment	searches the comment property
d, desc, description	searches the description property
a, app, application	searches the application property

Deleting Job Definitions

You can only delete job definitions when no other objects relate to them. For example, if the job definition is called in a job chain job, the job definition cannot be deleted until all job chain jobs using it have been deleted. You can see all job chain jobs that relate to the job definition in *Related Objects* in the lower detail pane and on the show page. Furthermore, jobs of a job definition will prevent you from deleting the job definition. These are found by selecting *Show Related Jobs* from the context menu. It is also possible to cancel and delete related jobs from the context menu.

The table in related objects contains three columns:

- Type - the type of object, like job for job chain job, Source for the source of the job with a link to it
- Related Object - the name of the object with a link to it

- Used As - objects can sometimes be used in different roles

Job definitions that are called in job chains will have the call-reference in *Related Objects*. The type will be *Job*, and the *Related Object* will be the name of the job. For example, if it is the first job in a step, it will be *Job 1*.

Security

The following privileges are available for job definitions:

Privilege	**Description**
JobDefinition.Create	Create job definitions
JobDefinition.Delete	Delete job definitions
JobDefinition.DeleteJob	Delete jobs
JobDefinition.Edit	Edit job definitions
JobDefinition.SubmitJob	Submit jobs
JobDefinition.View	Access job definitions

You can grant privileges on two levels, *Access* and *Admin;* a privilege granted on *Admin* level allows the grantee to grant the privilege to other users. These privileges can be granted *system-wide*, per *partition* or *isolation group*. Note that to be able to schedule a job, the user also needs privileges on the queue and other objects referenced by the job definition.

If you have the security module, which requires the *Module.Security* license key, you have an additional *Security* tab on the job definition. It allows you to specify which users can access, edit and delete the job definition.

- Granting and Revoking System Privileges
- Granting or Revoking Object Privileges

Editing XML Source

You can also edit the XML source directly or copy it into your editor of choice. This is especially handy when you want to perform repetitive tasks such as a *Search and Replace*. You enable the XML source tab in the user options by enabling the *Allow edit as XML* option; see Setting User Parameters. Special care needs to be taken when you edit in XML, as you will have to use entities for special signs, such as the *less-than* `(<)` sign, which needs to be replaced with `&amp;lt;`. Please see XML resources on-line for a list of common XML entities.

Procedure

To create a job definition:

1. Choose *Definitions* > *Job Definitions*.
2. Choose *New* from the context menu.
3. Fill in the below values.
4. Choose *Save and Close*.

Note

Some tabs require a new element to be created, this is done by choosing *New*.

Values

Topic	Relevant topic
Source Tab	Job source
Job Definition Tab	Job definition object
Parameters Tab	Job definition parameters
Parameter validation	Parameter validation
Wait Events Tab	Job definition wait events
Raise Events Tab	Job definition raise events
Job Locks Tab	Job definition job locks
Runtime Limits Tab	Job definition runtime limits
Restart Behavior Tab	Job definition restart behavior
Actions Tab	Job definition actions
Security Tab	Job definition object privileges

See Also

The Job Definition Types

Job Chains

Setting Job Definition Properties

Use

On the job definition tab of job definitions and job chains you can specify standard object properties like *Name* and *Application* as well as scheduling properties like *Default Queue* and *Priority*.

Naming and Location

Every job definition requires a unique name and should be part of an application. Applications are there to store your job definitions in containers along with other objects, that belong together. This allows you to migrate your processes or back them up easily with the Import/Export module.

Job Keep

Every job definition that is to be deleted automatically needs to have a keep clause. The keep clause defines how many jobs or how long jobs will be kept in the system. The default is to keep all jobs.

Scheduling Parameters

You can also set the default values of scheduling parameters, which are also available in the submit wizard.

- Default Queue - the queue that should be used by default
- Priority - the default priority of the job
- Resource - the resource of the job, will be used when selecting the process server in the queue, only process servers with the same resource are allowed to execute the job.

- Hold On Submit - the default value for the scheduling parameter *Hold on Submit* which defines the status of the job after the submit. Can be *true*, *false* (the default), or =<*expression*> in job chain jobs, which should be a boolean Redwood Expression Languageexpression.

When a job definition has missed its window for execution, it is set to overdue, you may configure the reaction here.

- *Act when jobs are overdue* - When a job has missed its window, change the status and/or raise an event.
  - *Set overdue jobs to status*
    - *Scheduled* - job will be executed the next time the window opens.
    - *Overdue* - marks the job as overdue, requires operator intervention.
    - *Canceling* - cancels the job, may require operator intervention.
    - *Held* - holds the job, requires operator intervention.
  - *Raise event for overdue jobs*

Format

Jobs can potentially generate output files. These can be *log files* that are guaranteed to contain log records in standard format, *text files* containing freely formatted information or error messages, or more complex files such as PDF or HTML files. Which files can be generated is dependent on the type of job that is executed.

The system will set the Format of the files created by the job.

Values

Field	**Description**	**Default Value**
Partition	The partition of the job definition	GLOBAL
Name	Name of the job definition	
Application	The name of the application	
Keep	The number of jobs to keep in history	All
Restrict to Time Window	The time window at which the job should run	
Time Zone	The time zone used to interpret the time window and/or submit frame	
Default Queue	The default queue of the job definition, the queue can be changed at submit time.	
Priority	The priority of the job, value from 1-100, 100 having the highest priority	
Resource	The resource of the job definition, used to determine on which process server the job will be run	
Default Output Format	The format of the job output files when the extension can not be mapped	Text
Default Log Format	The format of the job log files when the extension can not be mapped	Text
Hold On Submit	Should the job be submitted on hold by default?	No.

Maximum Restarts	If the job is automatically restarted, how many times may this happen	
Act when jobs are overdue	Should the system alert you when a job is outside of its time window because it could not start on time?	No
Set overdue jobs to status	The status an overdue job should be set to	
Raise event for overdue jobs	The event that is raised when the job is late.	
Returncode Map To Completed	The return code that represents a successful execution of the job, this field must contain all successful return codes if it is set, including 0.	0

Automatically Deleting Old Job Definitions

Use

The keep clause of a job definition can automate job deletion. The author of the job definition can set the keep clause to the desired number of jobs or the maximum age of jobs. If the number of jobs is set, this number can be counted per job owner, per key or overall for the system.

If you alter the keep clause of an existing job definition with existing jobs, then the conditions under which the existing jobs are kept is changed as well.

Changing the keep clause on a job definition that has many finished executions therefore results in massive work for the system. For example, when you run one million jobs of the same script without a keep clause and then set it to keep 10 per system, all one million finished jobs but 10 will be deleted.

Job Deletion

Jobs are not deleted immediately, but deleted by system jobs that run according to a submit frame every 15 minutes, by default. The job information in the central server is deleted by *System_ProcessKeepClauses* jobs, the output files are removed by *System_DeleteJobFiles* jobs. These are maintenance jobs that run in the background and are not displayed in the default *All (non-maintenance) jobs* job filter. If you need to monitor these jobs, you need to use the *Maintenance (background) jobs* job filter.

These job definitions run according to the built-in submit frames *System_ProcessKeepClausesSubmitFrame* and *System_DeleteJobFilesSubmitFrame*, you can change them to modify the interval at which the jobs run.

Maximum Age of Jobs

If you set a maximum age for a job, the system will fill in an expiration time stamp for the job when the job finishes. For example, if the keep clause is 1 day and the job finishes on 12 October at 11:02, then the expiration time stamp is set to 13 October at 11:02. When you edit the keep clause on the job definition, the changes affect all jobs of the job definition almost immediately.

Maximum Number of Jobs

If you set a maximum number of finished jobs, the system will keep track of the numbers of jobs per script. If you specify a number for the entire system, then an overall count is calculated and compared against the maximum. If you specify a number per user, then only the jobs ran for the user are compared to the set value. If you specify a number per key field, then the jobs with the same values for the parameters that are part of the primary key are considered. For example, if you run a job with a particular parameter `P_COMPANY_ID` which is set to be a key parameter, and the keep clause is 1 per key, then a run with `P_COMPANY_ID` set to 1 will not be deleted until another run with `P_COMPANY_ID` set to 1 is finished.

Editing the Job Definition Source

Use

Depending on the job definition type and your license, you can edit the source of the job definition.

The following job definition types allow you to specify a source, if the corresponding license key is enabled:

- BASH - For writing Bourne Again SHell scripts. BASH is installed on all Linux systems and optionally on all UNIX (compatible) platforms that SAP CPS runs on.
- CSH - For writing C Shell scripts. Available on most UNIX systems.
- KSH - For writing Korn Shell, for job definitions that run on UNIX (compatible) servers.
- PERL - For writing Perl scripts.
- PERL (Unicode) - For writing Perl scripts with UNICODE characters in your script source.
- CMD - For writing Microsoft Windows Command scripts.
- PS1 - For writing Microsoft Windows PowerShell language scripts. PowerShell is available as a download for Microsoft Server 2003, XP and Vista and is delivered with Microsoft Server 2008.
- VBScript - For writing Microsoft Windows Visual Basic Script job definitions.
- RedwoodScript - The RedwoodScript job definition type allows you to write server based scripts that interact with the repository using the RedwoodScript language.
- JCL_FTP - For writing JCL scripts submitted via FTP to remote IBM zSeries servers running the MVS or z/OS operating system controlled by JES2 or JES3.
- JCL - For writing JCL scripts submitted to IBM zSeries servers via a Redwood native agent.

Library

The library allows you to use your classes across all job definitions, which is especially useful for RedwoodScript job definitions. It can also be used for parameter validation, you need to setup Redwood Expression Entry points in the library for the methods you would like to expose to Redwood Expression Language. See the Redwood Expression Language for more information.

Run As User

Some job definition types allow you to specify a user under which the code in the job definition is executed, the user needs to exist and not be locked for the code to be executed on the target system.

SAP recommends the use of Credentials as you only have one object to change when the password changes, for example. When you use credentials there must be a credential for the user or virtual user on the node, also known as endpoint, where the job is running. See the Assign Users from External Platforms to Jobs with Credentials section for more information.

Values

The source tab contains the following fields:

Field	Description	Default Value
Type	The job definition type	

Library	The library containing your JAVA classes	Custom
Run As User	The user that will execute the source on the process server	
Source	The code to be executed	

Exceptions

The following job definition types do not allow you to change/edit the source code:

- JOBCHAIN
- JXBP
- SAPPI
- SAPR3
- SYSTEM

Defining Job Parameters

Use

Parameters are used to store values, which can optionally be changed at submit time. In job chains, parameters can also be shared and updated between jobs.

The following properties define the parameters:

- Name - is used in the source in job chains to refer to the parameter, you should only use alphanumeric characters and the underscore _ in names
- Direction - defines the way how this parameter is to be used.
  - In - value is provided in the job definition or the submit wizard and does not change
  - Out - value is provided by the job, at runtime
  - In/Out - value is provided in the job definition or the submit wizard and can be changed by the job at runtime
- Type - the data type of the parameter, you cannot store alphabetical characters in number parameters, for example
  - Character - parameter can only hold one single character
  - String - parameter can hold one or more alphanumeric characters
  - Number - parameter can only hold numbers
- Default Expression - the value of the parameter, can be a literal, like `Hello World` or an expression, like `=Time.now('Europe/Berlin')`.

The following properties specify the appearance in the submit wizard

- Display Order - defines the order in which parameters are evaluated as well as displayed in the submit wizard
- Group Name - allows you to create tabs in the submit wizard to group parameters
- Display - parameter is displayed in the submit wizard

Other properties

- Optional - parameter value can be null
- Part Of Key - parameter is used in keep clauses, see Automatically Deleting Old Job Definitions.

Values

Field	**Description**	**Default Value**
Name	the name of the parameter	
Direction	The direction can be in, out, or in/out. When a parameter is an *in* parameter, its value is read-only, if the parameter is *out* it does not have a default value, *in/out* parameters are read/write parameters.	
Type	The type of parameter to use, *Character* parameters can only be filled by a single character, *String* can be filled by any number of characters and/or numbers.	
Display Order	The order in which the parameters are displayed in the submit wizard.	
Group Name	Allows you to group parameters in tabs in the submit wizard	
Default Value Editable	Can the value be edited in the submit wizard?	[none]
Default Expression	The default value of the parameter, can be a literal or an expression.	
Simple Constraint Type	The type of constraint	None
Simple Constraint Data	A value used with the constraint, for example a table name with table constraints.	
Simple Constraint Message	Message that gets displayed when a value does not fulfill a constraint	
Optional	Can the parameter have a null value?	
Display	Should the parameter be displayed in the Submit wizard?	
Part Of Key	Should this parameter be used in the primary key? The primary key is used to calculate the expiry of a job, which once reached will cause the job to get deleted by a background job.	
Runtime	The value of the parameter is computed at runtime, once the job has been submitted. The value of the parameter will not show up in the submit wizard.	

Note

The *Display Order* of parameters is also the order in which parameters are resolved. If you use Redwood Expression Language to define a parameter based on the value of another parameter, the parameter in the expression must have a lower *Display Order* than the parameter it defines or the latter will not get resolved.

Example

Your job definition has two parameters, `param1` and `param2`. The data type of `param1` is *Number* and its *Default Expression* is set to `1`. The parameter `param2` is also a *Number* parameter, its *Default Expression* is set to `=parameters.param1 * 2`. Here, the *Display Order* is very important, if parameter `param2` has a lower *Display Order* than `param1` it will be displayed first. However, SAP CPS will not be able to resolve it and attempting to submit the job will result in an exception being thrown. This is due to the fact that `param1`, which is part of the *Default Expression* of `param2`, has not been evaluated, yet, when SAP CPS attempts to evaluate `param2`.

Parameter Validation using Constraints

Use

Constraints are widely used in databases to force valid values in tables, to make sure nobody makes a typo when entering data. In SAP CPS simple constraints are used to validate parameters.

There are different types of simple constraints:

- List - A list of valid values
- Pair List - A list of valid values with descriptions
- Expression - An expression
- Table - A list of values retrieved from a table
- Query Filter - A list of values retrieved with a filter, such as job filters

The *Simple Constraint Data* field is used to specify the list, expression, or table name. The *Simple Constraint Message* is displayed when the value of the parameter does not match the simple constraint. A good *Simple Constraint Message* could be the following:

```
"Please choose a valid value from the list, the value you entered is not valid."
```

All constraint types allow the use of variables:

- ${value} – value specified in the submit wizard.
- ${legalValues} – short list of legal values (This variable is empty when used in expression constraints)
- ${parameter} – name of the parameter.

List Constraints

The *List* constraint is the easiest to create and parameter-specific. The following illustrates a valid *List* constraint with the following in the *Simple Constraint Data* field:

```
A,B,C,D,E
```

Pair List

Like the *List* constraint, the *Pair list* constraint is the easy to create and parameter-specific. The following illustrates a valid *Pair List* constraint with the following in the *Simple Constraint Data* field:

```
A="First Letter",B="Second Letter",C="Third Letter",Z="Last Letter"
```

Note that the letters (A, B, C, and Z) are the values passed on to the parameter. The submit wizard will create a list of values for the user to choose from.

Expression

Like the *List* constraint, the *Expression* constraint is parameter-specific and consists of a Redwood Expression Language expression. The result of the expression is used to validate the value in the submit wizard.

An example message you can use is:

```
The value ${value} you specified is invalid for parameter ${parameter}, Valid values are ${legalValues}.
```

Using Redwood Expression Language expressions:

Use the keyword `value` for the actual value, in the following example force the value to be a number between 5 and 10.

```
=value >= 5 && value <= 10
```

You can also use the special constraint functions:

- Constraint.listConstraint(titles, list, bool)
- Constraint.pairListConstraint(titles, list, bool)

In the above functions, the following parameters can be used:

- titles is a comma separated list of titles.
- list is either a normal list or a pair list
- bool is a boolean expression involving the parameter value, `true` evaluates to *accept* and `false` to *reject*.

```
=Constraint.listConstraint('Country', '31,33,44', \
Table.getColumnString('Countries', value, 'Translation') !== '')
```

This example works with the following table named countries:

Key	Name	Abbreviation	Translation	Conversion_Rate
31	The Netherlands	NL	Nederland	1.0
33	France	Fr	France	1.0
49	Germany	De	Deutschland	1.0
502	Guatemala	Gt	Guatemala	11.6

Note that the column we retrieve in the boolean expression, *Translation* in the example above, is not really relevant. The value returned is compared to `''`, which means *empty*. If the column is not mandatory and is not filled, the operator will not be able to use the `value` associated with the column.

Table

Unlike the other simple constraints, the *Table* constraint allows you to specify a list of values once that can be used in multiple job definitions. The list of values can be maintained from one central location. The *Table* constraint requires you to fill a table. You can use the built-in *System_Variables* table or create your own. The value used will be the value from the *Key* column in the table. In the example above, the values available would be `31`, `33`, `49`, and `502`.

To use a table, you select *Table* in the *Constraint Type* field and fill the name of the table in the *Simple Constraint Data* Field.

Query Filter

The query filter constraint allows you to use object filters to generate list of values for parameters, this is especially handy for RedwoodScript job definitions. The submitter must be able to see the object filter, or the parameter value will not be evaluated at submit-time. If the parameter is optional, this has no major consequences, however, if the parameter is mandatory, the user will not be allowed to submit the job. Object filters also display a list according to the submitter's privileges, so two different users might see different lists.

In the following example RedwoodScript code, the job definition has a parameter named *Parameter1* which has a query filter constraint on the filter `All job Definitions`. When the job definition is submitted, you can choose a job definition for *Parameter1*, the value is a string, which can then be used to retrieve the job definition object, via its business key, for example.

```
{
  // Print the actual parameter value
  jcsOut.println("parameter=" + Parameter1);
  // Get a business key for the parameter value
  BusinessKey key = BusinessKey.valueOf(Parameter1);
  // Use the business key to retrieve the job definition
  JobDefinition jd = jcsSession.createBusinessKeyResolver().getJobDefinitionByKey(key);
  // Print the name of the job definition
  jcsOut.println("job definition=" + jd.getName());
}
```

The value of the query filter is `<object_type>:<partition>.<object_name>`, such as `JobDefinition:GLOBAL.System_Info` or `EventDefinition:GLOBAL.DataLoadComplete`, for example.

Values

Field	Description	Default Value
Simple Constraint Type	The type of constraint to use	None
Simple Constraint Data	The data to be used by the constraint (list, expression)	
Simple Constraint Message	The message to display when the constraint is violated	

See Also

- Privileges Required to use QueryFilters

Using Tables to Store Frequently Used Values

Use

Many complex process consist of jobs the work with the same parameter values, for instance:

- The name of an external system
- The processing day (or business day) which may not be the same as today
- The name of an SAP instance or client to use for a particular task

These values tend to be used by a large number of jobs, and change rarely. They might change:

- On a daily basis (for the business day)
- Between the Development, Quality Assurance and Production environments (for external system names).

In some cases there may be a pre-defined set of values from which the user can choose:

- A list of countries, provinces or cities
- A list of departments or cost codes

In these cases it is also useful to be able to validate the value a user has entered to ensure it is correct, and to prompt them with a list of valid values. This means that validation can occur at job submission time, rather than at runtime.

- Tables - Tables allow you to store frequently used values and have these form a list in the submit wizard.
- Table definitions - Table definitions define tables, listing valid columns and data types.

Actions

Tables and table definitions support the following actions:

Action	**Description**
Export	Export the table or table definition into a CAR file
Edit	Edit the table or table definition
Edit Security	Edit the security of the table or table definition
Delete	Delete the table or table definition
Duplicate	Make a copy of the table or table definition to create a similar one
New Table	Create a new table from the current table definition
Expand All	Expand all tables or table definitions in the current filter
New	Create a new table or table definition
Filter > New Filter	Create a new table or table definition filter
Filter > Edit Filter	Edit current table or table definition filter
Filter > Delete	Delete current table or table definition filter

Finding Tables and Table Definitions

You can search for tables and table definitions using filters and the *Search Tables* or *Search Table Definitions* box on the Tables or Table Definitions tabs, respectively. These boxes are known as the *intelliSearch* boxes and located under your username on the top right-hand side of the user interface. Filters allow you to specify a list of objects with static criteria. IntelliSearch allows you to specify complex queries in a simple way using prefixes. Prefixes are used to specify which property you are searching in and have short and long syntaxes. For example, if you want to display all tables with the term *users* in the comment, you would use the search criteria as follows:

```
c:users
```

You can search more than one property, as follows:

```
c:users n:ORA
```

Note

No spaces should be entered before or after the colon (*:*).

See the Advanced Object Search for more information.

The following table illustrates the available prefixes for tables and table definitions:

Prefixes	**Description**
n, name	searches the name property
c, com, comment	searches the comment property
d, desc, description	searches the description property
a, app, application	searches the application property

Deleting Table Definitions and Tables

You can only delete table definitions and tables when no other objects relate to them. For example, if there are job definitions that use the table, the table cannot be deleted until all job definitions that use it have been modified. You can see all job definitions that relate to the table in *Related Objects* in the lower detail pane and on the show page. You cannot delete table definitions if there is a table using it.

The table in related objects contains three columns:

- Type - the type of object with a link to it
- Related Object - the name of the object with a link to it
- Used As - objects can sometimes be used in different roles

Security

Privilege	**Description**
TableDefinition.Create	Create table definitions
TableDefinition.Delete	Delete table definitions
TableDefinition.Edit	Edit table definitions
TableDefinition.View	Access table definitions
Table.Create	Create tables
Table.Delete	Delete tables
Table.Edit	Edit tables
Table.View	Access tables

You can grant privileges on two levels, *Access* and *Admin;* a privilege granted on *Admin* level allows the grantee to grant the privilege to other users. These privileges can be granted *system-wide*, per *partition* or *isolation group*.

If you have the security module, which requires the *Module.Security* license key, you have an additional *Security* tab on the table and table definition. It allows you to specify which users can access, edit, and delete the table and table definition.

- Granting and Revoking System Privileges
- Granting or Revoking Object Privileges

Procedure

Create a table definition

1. Navigate to *Scripting* > *Table Definitions*.
2. Choose *New* from the context menu and enter a name and application (optional).

3. Choose the *Columns* tab, and fill in some columns, with data types and specify if the value is allowed to be `null`.

4. Choose *Save & Close*.

Create a table

1. Navigate to *Scripting* > *Table Definitions*.

2. Choose *New Table* from the context menu of a table definition and enter a *Name*, and *Application* (optional).

3. Choose the *Values* tab, and fill in some columns, the column names must exist in the table definition you specified. The *Key* column is the field that should contain the data that you want to use.

4. Choose *Save & Close*.

Use a table as a reference in a parameter for parameter values

1. Navigate to *Definitions* > *Job Definitions*

2. Choose *Edit* from the context menu of the job definition

3. Choose the *Parameters* tab and select the parameter you would like to use

4. In the *Simple Constraint Type* field choose *table*, enter the name of the table into the *Simple Constraint Type* field.

5. Choose *Save & Close*.

Result

When you submit the job definition, records in the table are available as list-of-values in the parameter field of the job definition.

Jobs Waiting on Events

Use

Jobs can wait for one or more events, this forms the basis of event-driven process scheduling. You specify as many events as you need to be waited on. The jobs based upon those definitions cannot be executed before the events upon which they are dependent are raised. When a job is created for a definition, it inherits all the events to be raised, and must wait for those.

A job waits only once for a specific event. So if the event was raised but the job did not start (for example, due to the closing of a job queue), the job does not wait again for an event from the same event definition the next time the job is released.

When a job waits for an event it gets the *EventWait* status.

Automatically started jobs

A event-waiting job can be auto-started when the event is raised. Make sure that all job parameters (if any) have default parameter values. You can set auto-start to:

- Never - A job will never be submitted.

- Always - A job will always be submitted.

- Only one - Only submit a job when there is no job of the same job definition in status *New*, *Initial*, or *Pending*.

Note

An event-waiting job must first be scheduled before it gets the *EventWait* status.

Clearing events

The job that is triggered by the event should clear the event, so that another event-waiter (created later) will wait for another occurrence of the event.

Values

Tab	Field	Description	Default Value
Wait Events	Event Definition	The event definition the job will be waiting for.	
Wait Events	Auto Submit	Should the job start automatically each time this event is raised?	Never.
Wait Events	Clears Event	Should the start of the job clear the event?	No.
Wait Events	Clear Comment		

Jobs Raising Events

Use

Jobs can be used to raise events when they reach a specific job status, this allows you to trigger other jobs.

When a job raises an event, the *Raiser Comment* field contains information on the raiser, this can be further customized.

The default raiser comment is the following:

`Event raised by Job:${jobid} (${jobdescription}) going to status ${jobstatus}`.

The following substitution parameters are available for raiser comments in job definitions and job chains:

- ${jobid} - The job id of the raiser job
- ${jobdescription} - the raiser job name, as displayed by the job monitor at the time of raising the event.
- ${jobstatus} - The status of the job at the time it raised the event.

Values

Tab	Field	Description	Default Value
Raise Events	Event Definition	The event definition the job will raise once it has finished.	
Raise Events	Raise Comment	The comment the raised event will have.	Event raised by Job:${jobid} (${jobdescription}) going to status ${jobstatus}.
Raise Events	Status To Raise On	The job status that makes the job raise the event once it reaches that state.	Completed
Raise Events	Error Code	The error code of the job.	

Jobs with Locks

Use

Locks are used to prevent jobs running simultaneously. A job that is attached to a lock can only start running when no other job is *holding* the lock. As soon as a job starts running, it *holds* the lock until it has reached a final job state.

Values

Field	Description	Default Value
Job Lock	The name of the lock the job will take and wait upon	

Job Runtime Limits

Use

Runtime limits ensure that a job runs for a predefined period of time or does not exceed a limit. Multiple limits are allowed with either minimal and maximal values.

The limits can be hard coded or you can use an offset of runtime estimation based on previous runs of the job.

Values

Field	Description	Default Value
Name	The name of the runtime limit	
Max Min Type	The type of runtime limit, *Maximum* and *Minimum* are allowed.	Maximum
Value	The value of the runtime limit, it can be fixed, or time-based or an offset in percent	
Kill Job	Should the job that has reached the limit be killed?	
Event Definition	Event to raise when the job has reached the limit.	

Restart Behavior

Use

Restart behavior is used to react to the final job status of the job, usually to break or continue the submit frame when something unexpected happens. When you break the submit frame, the job definition or job chain does not get submitted again according to the submit frame. By default, a *Completed* job definition or job chain will continue to get submitted according to the submit frame.

The following job states can be reacted on:

- *Completed*
- *Error*
- *Killed*
- *Canceled*
- *Unknown*

For each state one of the following reactions are available:

- *Continue Submit Frame* - The job is resubmit according to the submit frame.
- *Stop Submit Frame* - The job is not resubmit according to the submit frame.
- *Request Restart* - An operator message is sent requesting the restart, the operator needs to acknowledge and choose a restart option.
- *Restart* - The job is restarted.
- *Restart No Resubmit* - The job is restarted, however, it will not be resubmit according to the submit frame.

Additionally, you can always send an operator message.

Global and Partition-Wide Restart Behavior

The *Restart Behavior* can be set on the job definition, partition-wide, and system-wide. To set the *Restart Behavior* partition or system-wide, create a job chain named as follows:

- `<partition>.System_Defaults_Partition` - governs the *Restart Behavior* for the <partition> in which the job chain was created
- `GLOBAL.System_Defaults_System` - governs system-wide *Restart Behavior*

Note

The job chain will silently be turned into a special *System* Job Definition and be accessible in *Definitions > Job Definitions*. The only settings you are allowed to changed on these special job definitions are on the *Security* and *Restart Behavior* tabs.

The system processes the restart behavior in the following order, the first match is used:

1. Job Definition of job.
2. <Partition of Job Definition>.System_Defaults_Partition
3. GLOBAL.System_Defaults_System
4. Built-in defaults

Please see Controlling Global and Partition Restart Behavior for more information on defining the job definitions for system-wide and partition-level restart behavior.

Values

Tab	Field	Description	Default Value
Restart Behavior	Status	The status that will trigger the restart of the job.	
Restart Behavior	Action	ContinueSubmitFrame: The restart will occur following the submit frame. StopSubmitFrame: The submit frame will be broken. RequestRestart: An operator message will be sent to request restart permission. Restart: Restart the job immediately. RestartNoResubmit: The job will be restarted, the restarted job will not be resubmitted.	
Restart Behavior	restart Delay	The delay between the end of the job and the restart.	0

Restart Behavior	Operator Message to send	The operator message to send	

Controlling Global and Partition Restart Behavior

Use

Restart behavior is used to react to the final job status of the job, see Restart Behavior for information on restart behavior. These settings can be specified globally and for a particular partition by creating special job definitions.

The job definition for the system-wide settings is named *System_Defaults_System* and resides in the *GLOBAL* partition.

The job definition for the partition-level settings is named *System_Defaults_Partition* and resides in the partition for which you want to specify the restart behavior.

The only properties that can be set on these special job definitions are:

- *Description* - can only be set at creation-time in the ui.
- *Comment* - can only be set at creation-time in the ui.
- *Restart Behavior*
- Object-level security

Tip

The easiest way to create these special job definitions is to create a job chain with the appropriate name, it will be changed into a *System* job definition as soon as you save it.

Note

These job definitions are special and you cannot submit them or modify any other properties. Their sole purpose is to allow you to specify restart behavior settings at various levels.

Procedure

Creating system-wide restart behavior settings:

1. Navigate to *Definitions* > *Job Chains*
2. Choose *Create* from the context menu
3. On the Job Definition tab, fill `System_Defaults_System` into the *Name* field. Make sure the *Partition* field is set to `GLOBAL`.
4. On the *Restart Behavior* tab, select your desired restart settings.
5. Choose *Save & Close*.

Creating partition-level restart behavior settings:

1. Navigate to *Definitions* > *Job Chains*
2. Choose *Create* from the context menu

3. On the Job Definition tab, fill `System_Defaults_Partition` into the *Name* field. Make sure the *Partition* field is set to the appropriate partition.

4. On the *Restart Behavior* tab, select your desired restart settings.

5. Choose *Save & Close*.

Example

You want killed jobs in the *Finance* partition to be restartable on request:

1. Navigate to *Definitions* > *Job Chains*

2. Choose *Create* from the context menu

3. On the Job Definition tab, fill `System_Defaults_Partition` into the *Name* field. Make sure the *Partition* field is set to *Finance*.

4. On the *Restart Behavior* tab, select *Request Restart* for the *On Killed* property.

5. Choose *Save & Close*.

You forgot to set the description at creation-time or would like to change the description of the job definition *System_Defaults_System*.

Navigate to *Scripting* > *Shell* and use the following code:

```
{
//Get the job definition
JobDefinition jobdef=jcsSession.getJobDefinitionByName("System_Defaults_System");
//Set the description
jobdef.setDescription("System-wide Restart Behavior");
jcsSession.persist();
}
```

You forgot to set the description at creation-time or would like to change the description of the job definition *System_Defaults_Partition* in the *Finance* partition:

Navigate to *Scripting* > *Shell* and use the following code:

```
{
//Get the partition and job definition
Partition mypart = jcsSession.getPartitionByName("Finance");
JobDefinition jobdef=jcsSession.getJobDefinitionByName(mypart, "System_Defaults_Partition");
//Set the description
jobdef.setDescription("Restart Behavior for the Finance partition");
jcsSession.persist();
}
```

Customizing Jobs with Specific Actions

Use

Actions allow you to interact with objects while these are doing something, the most common one being setting the final job status. With *Actions* you can change the status to `Completed` even if the job reached status `Error`. You may define a set of conditions that must be met for this change to take place. If you want to interact with jobs from many job definitions, it might be easier to use a trigger. See the Using Triggers and Using Actions sections of the documentation.

Note

Actions require RedwoodScript which in turn requires an Module.Scripting license key.

There are three different types of actions that fire at three different times in the life-cycle of a job. The following three actions are available:

- *On Change* - When the status of a job changes but has not reached Queued (but does not fire when the status changes to *Chained*)
- *Pre Running* - Before a job has started running
- *Post Running* - After a job has completed

Post Running is the only action that allows changing the final status of a job, it is therefore the most used action. The other actions allow you to change the queue, start time or even hold the job if a set of defined conditions are met.

You need to understand the concepts of job states to get a clear understanding of when triggers and actions fire, see the Job States section for more information on the subject.

All actions have a log file, which will be attached to the job if it is not empty. When an error occurs in an action, the job will go to status *Error*, this can be changed in status handlers.

The following illustrates the three *Actions* that are available and what they can be used for:

Since actions are dependent on the job state, you must be familiar with the various job states, see the Job States section for more information.

Note

You are *not* allowed to persist in any of these actions, that is done for you at the end.

On change

- Any change before *Queued* and *Dispatched*
  - Does not fire on *Chained* status
- Can change *scheduled start time*, *queue*, or hold the job.

Pre running

- During *Queued*, *Dispatched*, *EventWait* and *LockWait*
- Can change *scheduled start time*, *queue*, or hold the job.

Post running

- Any transition from *Running* into a final state:
  - *Error, Completed, Killed, Unknown, Canceled*
- Can change the final state.

All actions have access to the job changing status, the old and new statuses as well as a jcsSession. The changes that are allowed in the actions are action dependent. You cannot change the *scheduled start time* in a post-running action, for example, because the job has finished running.

The source of the actions is written in RedwoodScript which allows interacting with the repository. The above uses are only the most common - it is perfectly possible to submit another job in any of the actions or even cancel the job in question.

Contexts

The following predefined objects are available in job definition actions. These are defined in the *Source Stub*. You can reload the source tab from the context-menu to display them.

Before Job On Change

Object	**Class**	**Example Code**
jcsOutLog	com.redwood.scheduler.infrastructure.logging.api.Logger	jcsOutLog.warning("This is a warning!");
jcsOnChangeContext	com.redwood.scheduler.api.scripting.variables.PreExecutingActionScriptObject	jcsOnChangeContext.setFailJobOnError(false);
jcsSession	com.redwood.scheduler.api.model.SchedulerSession	jcsSession.getUserName()
jcsJob	com.redwood.scheduler.api.model.Job	jcsJob.hold();
jcsOut	java.io.PrintWriter	jcsOut.println("This text will be printed to a specific output file on the job.");

Before Job Pre Running

Object	**Class**	**Example Code**
jcsOutLog	com.redwood.scheduler.infrastructure.logging.api.Logger	jcsOutLog.info("This is an informational message!")
jcsSession	com.redwood.scheduler.api.model.SchedulerSession	jcsSession.getQueueByName("UNIX_Generic");
jcsJob	com.redwood.scheduler.api.model.Job	jcsJob.hold();
jcsPreRunningContext	com.redwood.scheduler.api.scripting.variables.PreExecutingActionScriptObject	
jcsOut	java.io.PrintWriter	jcsOut.println("This text will be printed to a specific output file on the job.");

Before Job Post Running

Object	**Class**	**Example Code**
jcsOutLog	com.redwood.scheduler.infrastructure.logging.api.Logger	jcsOutLog.debug("This is a debug message!")
jcsPostRunningContext	com.redwood.scheduler.api.scripting.variables.PostRunningActionScriptObject	
jcsSession	com.redwood.scheduler.api.model.SchedulerSession	
jcsJob	com.redwood.scheduler.api.model.Job	jcsJob.restart();

jcsOut	java.io.PrintWriter	jcsOut.println("This text will be printed to a specific output file on the job.");

Values

Tab	Field	Description	Default Value
Actions	Type	The type of action to create.	
Actions	Enabled	Enable/disable action	
Actions	Source	Source code of the actions in RedwoodScript	

Example

A multi-step job chain has a job in the first step that may fail, but the rest of the chain must finish to clean up. The status of the job chain should be *Completed* if all steps *Completed* and *Error* when the first step went into error. The status handler on the first step for status *Error* is *Continue*. The following action is used to check if the first step reached status *Error*, and if so, throws a RuntimeException to force the job chain into status *Error*.

```
{
  //Check if the first step failed, if it did,
  //throw an exception to put the Jobchain job into Error

  String stepName = "Run SystemBackup";
  String statusName = "Error";

  // Get an iterator over the child jobs of the job chain job
    for (Iterator it = jcsJob.getChildJobs(); it.hasNext();)
      {
         Job myJob = (Job) it.next();
         String realStepName = (String) myJob.getDescription();
         String realStatus = (String) myJob.getStatus().name();

         if (stepName.equals(realStepName) && statusName.equals(realStatus))
           {
              throw new RuntimeException();
           }
      }
}
```

See Also

- Redwood Script
- Trigger Examples
- Triggers

Editing Job Definitions Safely

Use

Special care needs to be taken when you edit job definitions that have already been submitted and have not yet started or job definitions that have been submitted with a submit frame. This is especially true when you edit the parameter list by adding non-Nullable parameters or by changing a Nullable parameter to non-Nullable. If a job is resubmitted by a submit frame and the parameters are not valid, for example when a non-Nullable parameter contains a null value, the job will be aborted and reach the *Modified* status, which will require human intervention.

Every time a job definition is submitted, a *slave* copy of the job definition in its current state is made and attached to the job. The user cannot interact with this *slave* copy. When you edit the *master* job definition before the job has reached the *Dispatched* status, the new job definition will be used. However, when a job definition is used several times in a submitted job chain, the new job definition will only be used when the first occurrence of the job chain job has not reached *Dispatched* at modification time. This has been implemented to allow job chains to remain consistent.

Substitution Parameters

Use

You use substitution parameters in specific objects to insert dynamic values, like a file name or a job id. Depending on the repository object, specific substitution parameters can be used.

Job Definitions

The following substitution parameters can be used in some job definitions:

- ${SystemId} - system id with all characters except A-Z, a-z and 0-9 replaced with underscores.
- ${date:format} - date using format 'format'. The format is any valid Java SimpleDateFormat.

Platform agent

The platform agent log file can contain a filename pattern, stderr, or - (for stderr). May contain any of the following substitution parameters:

- ${ProcessId} - Process id of the process
- ${ProcessType} - Process type (standalone, service, ...)
- ${ProcessName} - Process name (network-processor, ...)
- ${InstallDirectory} - install directory.
- ${TraceDirectory} - ${DataRootDirectory}/trc.
- ${DataRootDirectory} - Base directory for platform agent files.
- ${ProcessServer} - Name of the process server that the agent is configured for.
- ${FileSeparator} - Character(s) between directory and file (for example / on Unix and \ on Windows).

Events

The following substitution parameters are available for raiser comments in job definitions and job chains:

- ${jobid} - The job id of the raiser job
- ${jobdescription} - the raiser job name, as displayed by the job monitor at the time of raising the event.
- ${jobstatus} - The status of the job at the time it raised the event.

The following substitution parameters are available for raiser comments of file events:

- ${CurrentDateTime} - Current date & time
- ${CurrentTimeStamp} - Current time in epoch

- ${FileDateTime} - File modification date & time
- ${FileTimeStamp} - File modification time in epoch
- ${ServerDateTime} - Central server date & time
- ${ServerTimeStamp} - Central server time in epoch
- ${server} - Platform agent name
- ${filename} - file name

For the above, you can specify a JavaDateFormat for the `*DateTime` variables as follows:

```
${FileDateTime:HHmmss}
```

The following substitution parameters are available for the *Move Directory* field of file events:

- ${Name} - The base filename plus extension of the detected file.
- ${BaseName} - The base filename of the detected file.
- ${Extension} - The extension of the detected file.
- ${Dot} - A dot if the detected file had an extension, and nothing otherwise.
- ${TimeStamp} - The modification timestamp that the detected file has, in a numeric format containing the milliseconds since 1970, usually referred to as epoch or Unix time.
- ${DateTime} - The modification timestamp that the detected file has, in a string format `YYYYMMDDHHMMSS`.

Queries

The *columnFormat* parameters of the Table.formatRow(table, key, rowStart, columnFormat, columnSeparator, rowEnd) function takes the following substitution parameters:

- *{0}* - column header
- *{1}* - value
- *{2}* - columnSeparator (should always be the first)

Specifying Dates Using Java Date Format

Use

The Java SimpleDateFormat class is used to format dates in this case. An example for a dynamic date using Java SimpleDateFormat would be the following:

${Time.now:'yyyyMMdd'}

The following table lists alls valid letters that can be used to form a pattern which represents the date format.

Letter	Date or Time Component	Presentation	Examples
G	Era designator	Text	AD
y	Year	Year	1996; 96

M	Month in year	Month	July; Jul; 07
w	Week in year	Number	27
W	Week in month	Number	2
D	Day in year	Number	189
d	Day in month	Number	10
F	Day of week in month	Number	2
E	Day in week	Text	Tuesday; Tue
a	Am/pm marker	Text	PM
H	Hour in day (0-23)	Number	0
k	Hour in day (1-24)	Number	24
K	Hour in am/pm (0-11)	Number	0
h	Hour in am/pm (1-12)	Number	12
m	Minute in hour	Number	30
s	Second in minute	Number	55
S	Millisecond	Number	978
z	Time zone	General time zone	Pacific Standard Time; PST; GMT-08:00
Z	Time zone	RFC 822 time zone	-0800

Example

The given date and time are 2008-07-04 12:08:56 local time in the U.S. Pacific Time time zone.

Java SimpleDateFormat	**Result**
${now:"yyyyMMddHHmm"}	200807041208
${now:"yyyy.MM.dd.HH-mm-ss"}	2008.07.04.12-08-56
${now:"EEE-dd-MM-yy"}	Wed-04-07-08
${now:"hmma"}	1208PM

Redwood Job Definition Types

Use

Job definition types define a type of job definition and the service required to execute the jobs of that type. Job definition types can be assigned to a job definition or to a process server. When job definition types are assigned to a process server, they define the type of jobs that the process server can run and services that are needed. You must only submit job definitions of a certain job definition type into queues that are handled by process servers capable of running those job definition types, or those jobs can never be started.

When you create or edit a process server, the required services are added automatically to the process server based on the job definition types you configured, so there is no need to change services on a process server.

Job definition types are part of the *Configuration* section in the navigation bar.

Job definition types only support the basic action *Refresh*. It is not possible to *Create*, *Edit* or *Delete* job definition types.

To view a job definition type, choose a job definition type in the overview table. The job definition type details are shown in the lower detail pane.

The following standard job definition types are provided with SAP CPS:

Redwood System Job Definition Types

- RedwoodScript - The RedwoodScript job definition type allows you to write server based scripts that interact with the repository using the RedwoodScript language.
- Mail - The Mail job definition type allows you to send custom emails.

The following are also strictly speaking job definition types, but do not allow you to write your own script source code:

- JobChain - You can graphically compose ordered sequences of parallel jobs.
- JXBP - To run SAP Java jobs on the Java Scheduler via the Java eXternal Batch Processing (JXBP) interface.
- SAPPI - To control communication channels in SAP systems via the SAP PI interface.
- SAPR3 - To run SAP ABAP jobs accessible via various ABAP interfaces such as XBP and BW-SCH.
- System - System jobs provided by Redwood.

Note

Some job definitions are part of a module and require a specific license, please check the License section for more information.

Platform Job Definition Types

Platform Job Definition Types are available to connect SAP CPS to external systems such as Windows or Unix servers.

Note

Please pay attention when you mix Redwood job definition types with platform job definition types, see Process Server Services for more information.

Finding Job Definition Types

You can search for job definition types using filters and the *Search Job Definition Types* box on the Job Definition Types tab. This box is known as the *intelliSearch* box and located under your username on the top right-hand side of the user interface. Filters allow you to specify a list of objects with static criteria. IntelliSearch allows you to specify complex queries in a simple way using prefixes. Prefixes are used to specify which property you are searching in and have short and long syntaxes. For example, if you want to display all job definition types with the term *dev* in the comment, you would use the search criteria as follows:

```
c:dev
```

You can search more than one property, as follows:

```
c:dev n:ORA
```

Note

No spaces should be entered before or after the colon (:).

See the Advanced Object Search for more information.

The following table illustrates the available prefixes for job definition types:

Prefixes	Description
n, name	searches the name property
c, com, comment	searches the comment property
d, desc, description	searches the description property
a, app, application	searches the application property

Values

Property	Description
Name	The name of the job definition type.
Description	An optional description of the job definition type.
Comment	An optional comment for the job definition type.
Process Server Job Definition Types	A list of process servers that support this job definition type.
Handler Service	The service needed to process jobs of this job definition type.

Using the RedwoodScript Job Definition Type

Use

Note

The RedwoodScript job definition type is available with the Scripting module, which requires a specific license.

RedwoodScript is based on the Java language. It allows you to create job definitions with RedwoodScript in its source. The RedwoodScript job definition type is used to to create jobs that interact primarily with the server interfaces themselves. For instance, you can write a job that submits other jobs or retrieves and combines meta data or job output from other jobs.

Note

You must assign at least one process server to run RedwoodScript job definitions in order to use the job definition type.

Variable and Parameters

Variables that you require to hold intermediary values are defined in the same way as you would normally define variables in Java, using the standard syntax `var myVar` or `var myVar = "initial value"`.

Job definition parameters are available directly in RedwoodScript as Java variables, as shown in the examples below.

Note

When a job reaches status *Error*, the *Out* values of parameters are not always set. You can use a *Post Running* action to set them if required.

Examples

The following code shows how to access the job's input parameter names and values:

```
{
  jcsOutLog.debug("Job parameters");
  for (Iterator it = jcsJob.getJobParameters(); it.hasNext(); )
  {
    JobParameter par = (JobParameter) it.next();
    jcsOutLog.debug(par.getJobDefinitionParameter().getName() + "=" + par.getInValue());
  }
}
```

Within a job chain you can not only access the job parameters, but also the job chain parameters -- if needed. The following code shows how to do this, by creating an object that is the grandparent of the current job. The parent of the job is the 'step' within the job chain, and the grandparent is the job for the job chain itself. If you need to access further ancestors the method would be to call `getParentJob()` recursively. The following code can also be used to execute code depending on the depth of the job in the chain, or if the job is part of a chain or not.

```
{
// Does this job have a step  - which would imply that we are part of a job chain
if (jcsJob.getJobChainStep() != null)
{
  JobChain chain =  jcsJob.getJobChainStep().getJobChain();
  if (chain != null)
  {
    Job chainJob = jcsJob.getParentJob().getParentJob();
    // We are in a chain, our parent's parent is the chain job
    // Since our parent is the step job, the step's parent is the job chain job.
    jcsOutLog.debug("Part of a chain " + chain.getJobDefinition().getName());
    jcsOutLog.debug("Chain parameters");
    for (Iterator it = chainJob.getJobParameters(); it.hasNext(); )
    {
      JobParameter par = (JobParameter) it.next();
      jcsOutLog.debug(par.getJobDefinitionParameter().getName() + "=" + par.getInValue());
    }
  }
}
else
{
  jcsOutLog.debug("Not in a chain");
}
}
```

More examples are available in the Using RedwoodScript in Jobs section.

Using the Mail Job Definition Type

Use

The mail job definition type allows you to create job definitions that send email.

Note

Use of the Mail module requires the *Module.Mail* key in your license.

Note

The notification and alerting features of the Mail module require you to configure every job chain that is to use these features. If you make extensive use of alerts and/or notifications, Redwood recommends the *Active Alerting* module, as you can configure alerts and notifications centrally. Please see the Different Versions section for more information.

You can use the *System_Mail_Send* job definition to send simple mails, but as soon as you want to customize the email, you can use the Mail Job Definition Type.

The job definition type allows the use substitution parameters and allows you to change the content type of the email.

The Source of a mail job definition is composed of sections, which contain `<variable>=<value>` pairs. The default section is the body of the email and has no header; the following other optional sections can be used:

- *[Attachments]* - File attachments are specified here
- *[Headers]* - For email headers
- *[Body]* - Metadata about the body, like content-type and character-set (charset).

Blank lines and comments (lines starting with #) are ignored in the sections above. The contents of the default section is assumed to be the body of the email. If the default section starts with <, then the content type is set to HTML, you can override this in the [body] section.

You can use the following substitution parameters:

- *${jobname}* - The description of the job that sent the email.
- *${jobid}* - The job id of the job that sent the email.
- *${<parameter_name>}* - The value of the parameter of the email job, for example, if the email job has a parameter `Subject` it would be referenced as `${Subject}`. This allows you to use Redwood Expression Language in parameters and use the output of the functions in the mail body.

The following ant expressions can be used to generate the file name of attachments, especially when you use a GLOB expression (matching with '*' and '?') for the file name:

- *@{name}* - The name of the file to attach.
- *@{index}* - A simple counter.

Example

For the below example to work, the job definition needs to have a parameter `TimeNow`.

```
<html>
<body>
<h4>Latest financial results for board-members</h4>
<p>The financial results as of: ${TimeNow}</p>
<p>See attached <code>report</code> for a summary of the latest results.</p>
<br/>
<i>This message contains confidential and legally privileged information. In the event of any error in transm
```

```
immediately and not to disclose or make use of this information. No warranties or assurance are made or given
of this e-mail and attachments. No liability whatsoever is accepted for consequences arising from this e-mail
<i>This email was generated by ${jobname}:${jobid}.</i>
</body>
</html>
```

For the below example to work, the mail job definition needs parameters `Mail_Subject`, `Mail_To`, `TimeNow` and `InputFile`, the latter is the file that was being processed by the job chain (it must be available in `/var/reports/inbox/` and accessible to the mail module - see the AccessiblePath registry key). The job needs to be in at least the third step, the first job of the first step must fail (reach status *Error*) and have a *Final Status Handlers* set to *Continue* on *Error*, there must also be a job in the second step that produces output.

```
<html>
<body>
<h4>Job failed badly</h4>
<p>The time now is: ${TimeNow}</p>
<p>See attached <code>stderr</code> for details about the error.</p>
<i>This email was generated by ${jobname}:${jobid}.</i>
</body>
</html>

[Attachments]
#Attach stderr from job 1, step 1
error.txt=Step 1, Job 1:stderr
#Attach all files from Step2, job 1
@{name}-@{index}.txt=Step 1, Job 1:*
#Attach a file from the server
Attachment.doc=/var/reports/inbox/${InputFile}

[Headers]
#set the reply-to address and override subject
reply-to=it@masalan.com
subject=Something went wrong, original subject: ${Mail_Subject}

[Body]
Content-Type=text/html; Charset=UTF-8
```

See Also

- Basic Notification and Alerting with the Mail Module
- Configuring Mail and Sending Emails
- Customizing Emails with the Mail Module

Creating Job Chains

Use

A job chain executes one or more job definitions (job definition jobs) in parallel or sequentially. The job definition jobs are grouped into steps. All jobs in a step are executed simultaneously (parallel processing) and the process server waits until all jobs in a step get a final job status before proceeding with the next step (sequential processing). Jobs executed by job chains are still subject to routing using queues and resources and the limitations applied by queue limits and Locks.

A job chain is also a job definition of type JOBCHAIN, where the scheduling and runtime behavior of the chain are defined.

Actions

Job definitions have the following actions:

Action	Description
Show job chain diagram	Display a diagram of the job chain with related objects
Submit	Submit the job chain
Cancel non-final state jobs	Cancel jobs that have not reached a final state, see the Job States section for more information on job states.
Delete final state jobs	Delete jobs that have reached a final state, see the Job States section for more information on job states.
Monitor related jobs	Jump to the job monitor with a temporary job filter displaying jobs for the job chain
Reset Statistics	Reset the runtime statistics for the job chain
Export	Export the job chain
Edit	Edit the job chain
Edit Security	Edit the security of the job chain
Delete	Delete the job chain
Duplicate	Duplicate the job chain
Expand all	Allows you to expand the entire tree of the job chain
New	Create a new job chain
Filter > New Filter	Create a new job chain filter
Filter > Edit Filter	Edit an existing job chain filter
Filter > Delete Filter	Delete a job chain filter

A step gets an error status when one of its job definition jobs runs into a erroneous status. A job chain aborts by default when one of its steps gets an error. This can be overridden by defining a *status handler*.

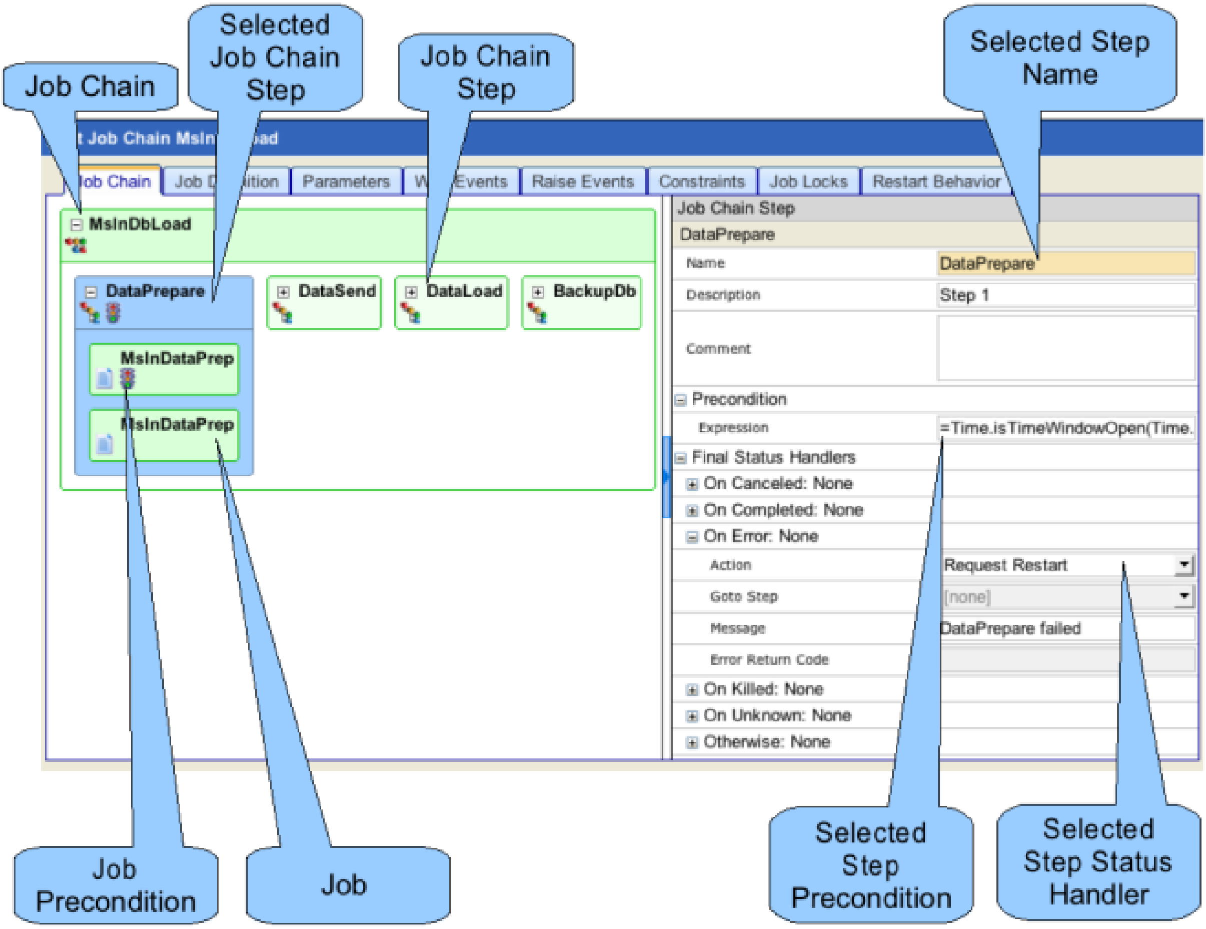
Status handlers can be defined at two levels:

- Job chain level, called default status handlers.
- Step level, called step status handlers.

A status handler defined at job chain level acts as default for a step. Job chain status handlers are overruled by any triggered step status handler. A status handler is triggered when one or more jobs in a step or in the job chain get a specific job status:

- An 'Other' handler will always trigger (even if there aren't any jobs), so it should always come last
- A 'Completed' handler triggers only if there is not a single non-Completed job
- A 'Cancelled/Error/Killed/Unknown' handler triggers when there is at least one job chain job with that status.

Restart behavior is used to break or continue the submit frame when something unexpected happens. When you break the submit frame, the job definition or job chain does not get submitted again according to the submit frame. By default, a *Completed* job definition or job chain will continue to get submitted according to the submit frame.



Job Chain Step

DataPrepare

Name	DataPrepare
Description	Step 1
Comment	
⊟ Precondition	
Expression	=Time.isTimeWindowOpen(Time.
⊟ Final Status Handlers	
⊞ On Canceled: None	
⊞ On Completed: None	
⊟ On Error: None	
Action	Request Restart
Goto Step	[none]
Message	DataPrepare failed
Error Return Code	
⊞ On Killed: None	
⊞ On Unknown: None	
⊞ Otherwise: None	

This is the main window of the job chain editor. From here the structure of the job chain can be edited, and other tabs can be accessed that allow you to change the properties of the various parts of the chain:

- the job chain itself: the job definition and the default status handlers.
- the steps: the step name and precondition function
- the jobs:
  - the job definition and precondition function
  - the parameter mappings
- testing

Most of actions can be accessed in several ways:

- the right-hand-side detail pane acts on the currently selected item.
- choosing the context menu of an item (the job chain name, the box of the job chain job, or the step name) shows the available actions.
- the links in the boxes give access to the job chain details or the job details.

You can leave the job chain details pane without saving or canceling the changes. They are stored in memory as long as the browser session exists, they are not committed to the database and have no effect when the job chain is submitted. However, do not edit the same job chain in more than one edit screen.

In the upper right of the job chain box is a white box with a plus (+) or a minus (-). The plus (+) is used to expand the view to show any child components of the part of the chain. The parent and all child components are shown. The minus (-) is used to collapse or hide the child components of the chain. Only the parent is shown.

The empty box with the dashed line outline represents an empty step in the job chain. This empty box is shown at the end of a chain. This is a place holder and shows the end of the chain. The empty box is also shown in the job chain flow when a step has been defined, but no job definitions have been added to the step.

Note

The changes are committed to the database only after you choose *Save* .

Special care needs to be taken, when you edit a job definition which is currently (re)scheduled; the new job definition will take effect on all jobs that have not reached the *Dispatched* job status at modification time.

Finding Job Chains

You can search for job chains using filters and the *Search Job Chains* box on the Job Chains tab. This box is known as the *intelliSearch* box and located under your username on the top right-hand side of the user interface. Filters allow you to specify a list of objects with static criteria. IntelliSearch allows you to specify complex queries in a simple way using prefixes. Prefixes are used to specify which property you are searching in and have short and long syntaxes. For example, if you want to display all job chains with the term *import* in the comment, you would use the search criteria as follows:

```
c:import
```

You can search more than one property, as follows:

```
c:import n:BI
```

Note

No spaces should be entered before or after the colon (*:*).

See the Advanced Object Search for more information.

The following table illustrates the available prefixes for job chains:

Prefixes	Description
n, name	searches the name property
c, com, comment	searches the comment property
d, desc, description	searches the description property
a, app, application	searches the application property

Deleting Job Chains

You can only delete job chains when no other objects relate to them. For example, if the job chain is still referred to in another job chain, for example, the job chain cannot be deleted until all relations have been deleted. You can see relations that relate to the job chain in *Related Objects* in the lower detail pane and on the show page. Furthermore, jobs of a job chain will prevent you from deleting the job chain, these will not be displayed in the *Related Objects* table.

The table in related objects contains three columns:

- Type - the type of object, like job for job chain job, Source for the source of the job with a link to it
- Related Object - the name of the object with a link to it
- Used As - objects can sometimes be used in different roles

For example, job chains that are called in other job chains, will have the call-reference in *Related Objects*. The type will be *Job*, and the *Related Object* will be the name of the job, for example, if it is the first job in a step, it will be *Job 1*.

Editing XML Source

You can also edit the XML source directly or copy it into your editor of choice. This is especially handy when you want to perform repetitive tasks such as a *Search and Replace*. You enable the XML source tab in the user options by enabling the *Allow edit as XML* option; see Setting User Parameters. Special care needs to be taken when you edit in XML, as you will have to use entities for special signs, such as the *less-than* `(<)` sign, which needs to be replaced with `&amp;lt;`. Please see XML resources on-line for a list of common XML entities. Also, ending spaces are stripped, when you edit in XML, make sure you do not leave trailing spaces or you will not be able to the object. See Editing Objects in XML for more information.

Procedure

The `Values` section below contains information about the various settings that are used in the following procedures:

- Creating Job Chains with Sequential Jobs
- Creating Job Chains with Parameter Handling
- Creating Job Chains with Parallel Jobs
- Creating Job Chains with Job Dependencies
- Creating Job Chains for multiple SAP Systems
- Managing Process Chains

Values

Tab	Relevant topic
Job Definition	Setting Job Definition Properties
Parameters	Defining Job Parameters
Parameter Validation	Parameter Validation
Wait Events	Jobs Waiting on Events
Raise Events	Jobs Raising Events
Job Locks	Jobs with Locks
Runtime Limits	Job Runtime Limits
Restart Behavior	Restart Behavior

Actions	Customizing Jobs with Specific Actions
Security	Object Privileges
Job Chain Steps	Job Chain Steps
Job Chain Jobs	Job Chain Jobs

Job Chain Steps

Use

Job chain steps are executed in sequence. They contain jobs that may be executed in parallel. Steps may also have precondition functions that control whether they are executed or not. Steps may also define their own status handling that controls what happens when a job goes into a final status that is not *Completed*.

The attributes of a step are:

- *Name* - name of the step
- *Description* - description of the step
- *Comment* - comment of the step
- *Precondition* - start condition (as an expression that should evaluate to `true` if the step should run)
- *Final Status Handlers* - the actions to take when all jobs have reached a final status

Precondition

Steps have a single, optional precondition function. A step with no precondition always executes. A step with a precondition only executes if the precondition returns true. If the precondition returns false, then the step is skipped.

The precondition is an expression in Redwood Expression Language.

Note

Preconditions are executed before the job is persisted, or saved in the database. This has the effect that it is not possible to access itself or, when it is a job chain, any other job in the same job chain.

Final Status Handlers

Final status handlers of job chain steps control the behavior of the step when all jobs in that step reach a final status. When this is the case, the step status handlers are evaluated. If the step does not define a matching handler, then the chain handlers are consulted.

Note

Step-level *Final Status Handlers* are not evaluated for disabled or skipped steps. If you want to use a precondition and *Final Status Handlers* (to match Otherwise), set the precondition on all job chain calls of the step instead of on the step.

Status handlers are evaluated as follows:

- An *Otherwise* handler will always trigger (even if there aren't any jobs)
- A 'Completed' handler triggers only if *all* jobs reached status *Completed*

- A 'Cancelled/Error/Killed/Unknown' handler triggers when there is at least one job chain job with that status.

If no status handler overrules the default, the default setting is followed.

The default setting is as follows:

- If all jobs of the step reached status *Completed* continue with next step or if this is the last step set the job chain to *Completed*.
- If any of the jobs of that step reached any other final state than *Complete*, set the step and parent job chain to *Error*.

To customize the behavior, the following actions are available:

- *Status* - The status the job chain, step or job must reach.
- *Action* - The action to take when the step reaches a given status can be:
  - *Goto Step* - go to another step within the job chain
  - *Raise Error* - raise an error by returning the `Return Code`
  - *Mark Completed* - change the status of the job chain to *Completed*
  - *Request Restart* - send operator message to choose a restart option and put the job to status *Console*
  - *Restart Step* - restart the current step
  - *Restart Chain* - restart the chain
  - *Continue* - continue with the processing of the chain
- *Message* - the error message to show when the action is set to *Raise Error*
- *Error Return Code* - the error code to raise when the action is set to *Raise Error*
- *Goto Step* - the step the *Goto Step* refers to.

The default processing order of final status handlers is as follows:

1. *Canceled* => On Cancel status handler
2. *Error* => On Error status handler
3. *Killed* => On Killed status handler
4. *Unknown* => On Unknown status handler

Otherwise is always last

The order of the status handlers can be changed via scripting, please note that any status handler coming after *Otherwise* will be ignored.

Job Chain Jobs

Use

A job in a job chain is a call to a job definition. It may also have an optional precondition function.

The attributes of the job itself are:

- *Job Definition* - The name of the job definition or job chain to be called.

If a job definition requires that parameters be specified, or has parameter values that must come from other jobs, then parameter mappings must also be supplied under *Input Parameters*.

Parameter values can refer to other parameters in the job chain, be set manually, or via an expression.

- *Reference* - Allows you to choose which job chain or job chain job parameter value to use for this parameter. The naming works in the following way: *{job chain name| step name:job number}:Parameter*, for example,
  - *ORA_DATALOAD:SID* - the value of the job chain level parameter *SID* of the current job chain called *ORA_DATALOAD*.
  - *DATALOAD:Job 3:SID* - the value of the parameter *SID* of the 3rd job in the DATALOAD step.
- *Expression* - the value of the parameter, can be a string, number, character, date time zone, or an expression in Redwood Expression Language.

Note

When a job reaches status *Error*, then *Out* values of parameters might not get set properly; this depends on the job definition type and the way you set the exit code, see the job definition type-specific topics for more information.

Preconditions

Jobs have a single, optional precondition function. A job with no precondition always executes. A job with a precondition only executes if the precondition returns true. If the precondition returns false, then the job is skipped.

The default input format for expressions is a string, in preconditions are usually made of functions, function expressions must start with an equals sign (=).

The precondition is an expression in Redwood Expression Language.

Note

Preconditions are executed before the job is persisted, or saved in the database. This has the effect that it is not possible to access itself or, when it is a job chain, any other job in the same job chain.

Scheduling Parameters

If a job in a job chain requires scheduling parameters to be specified then these parameters must also be set. This window allows for setting job chain job scheduling parameters.

The following scheduling parameters can be set:

- *HoldOnSubmit* - expression for hold on submit. Can be *true*, *false* (the default), or =<*expression*> which should be a boolean expression.
- *Priority* - Number from 1 to 100, the higher the number, the higher the priority.
- *Queue* - The name of the default queue.

Note

For SAP jobs, the queue defines which SAP instance the job will run against, when the *SAP System* parameter is not set, SAP jobs require the *Queue* scheduling parameter or the *SAP System* parameter be set as there is no default queue for SAP jobs.

Creating Job Chains with Sequential Jobs

Use

Job chains allow you to easily create dependencies between different jobs; reducing the time between two jobs that are to run sequentially to mere seconds. &sapjobchains;

Status Handlers

&sapccmsjobchains; Job chains have two levels of status handlers, on the step level and on the job chain level.

- When no status handler is found on the step level, the status handler on the job chain level is used.
- When there is none on the job chain level either, then the default behavior is used.

The default behavior for job chains is the following:

- When a step has reached status completed the following step is started or, when this is the final step the chain status is set to completed.
- When a step has reached status error the job chain status is set to error.
- When a step has been canceled the job chain status is set to error.

It is better to keep the amount of steps in a job chain to a minimum for performance reasons.

Procedure

1. Choose *Definitions > Job Chains* in the Navigation bar.
2. Choose *New* from the context menu.
3. Choose the *Job Definition* tab and fill in a name for the job chain.
4. On the *Job Chain* tab, choose the name of the job chain.
5. Choose *Add Step* from the context menu, notice that a new step with the caption *Step 1* has been added.
6. Choose *Add Job* from the context menu of *Step 1*.
7. Enter the part of the job definition name into the *Job Definition* field.
8. Choose the desired job definition out of the list of available job definitions.
9. Under *Scheduling*, add a parameter value for *Queue*.
10. Perform actions 5 to 10 of this procedure for each required step.
11. Choose *Save and Close*.

Example

Create a job chain with two steps, each step containing one job.

1. Choose *Definitions > Job Chains* in the Navigation bar.
2. Choose *New Job Chain* from the context-menu.
3. Choose the *Job Definition* tab enter *MSLN_AP_INV_IN* in the *Name* field.
4. Choose *New* next to the *Application* field, name it *MSLN_AP*.
5. Choose *MSLN_AP_INV_IN* on the *Job Chain* tab.
6. Choose *Add Step* from the context-menu.
7. Choose *Step 1*
8. Choose *Add Job* from the context-menu and choose the newly created box.
9. Enter *info* into the *Job Definition* field.
10. Choose System_Info from the list of available job definitions.
11. Fill *10* into the *Priority* field.
12. Choose *MSLN_AP_INV_IN*.
13. Choose *Add Step*.
14. Choose *Step 2*, choose *Add Job* from the context menu and then choose the newly created box.
15. Enter *Sleep* into the *Job Definition* field.
16. Choose *System_Sleep* from the list of available job definitions.
17. Enter the value *System* into the *Queue*; note that in this case, this task is optional. Since we did not define a default queue for the job chain, the submit wizard will require you to fill the queue field, this also means that you cannot submit this job chain automatically, from an event for example. You define a default queue for the job chain on the *Job Definition* tab, see the Creating Job Chains with Parameter Handling topic for an example.
18. Choose *Save & Close* to save the job chain.

Submit the job job chain

1. Navigate to *Definitions > Job Chains* in the Navigation bar.
2. Choose *Submit* from the context menu of the job chain 'MSLN_AP_INV_IN".
3. Choose *Next*.
4. Fill *System* into the *Queue* field, and choose *Submit*.

Creating Job Chains with Parameter Handling

Use

The SAP jobs of a job chain require parameters, there are two parameter classes, scheduling parameters and job parameters.

Scheduling parameters define the status of the job chain at submit time, the queue and the priority of the job.

For SAP jobs, the queue defines the SAP System where the job runs.

Job parameters have three directions, *IN*, *OUT* and *IN/OUT*.

Job chain parameters have the following levels:

- job chain level parameters, which are filled at submit time and can be **common** to one or more jobs.
- job level parameters, which are **individual**.

Note

The data type must always match when you map parameters. For example, most SAP type job definitions have string parameters.

Note

The *Display Order* of parameters is also the order in which parameters are resolved. If you use Redwood Expression Language to define a parameter based on the value of another parameter, the parameter in the expression must have a lower *Display Order* than the parameter it defines or the latter will not get resolved.

Procedure

1. Choose *Definitions* > *Job Chains* in the Navigation bar.
2. Choose *New* from the context menu.
3. Choose the *Job Definition* tab and fill in a name for the job chain, choose *Save*.
4. On the *Job Chain* tab, choose the name of the job chain.
5. Choose *Add Step* from the context menu, notice that a new step with the caption *Step 1* has been added.
6. Choose *Add Job* from the context menu of *Step 1*.
7. Choose *Job 1* and enter the part of the job definition name into the *Job Definition* field.
8. Choose the desired job definition out of the list of available job definitions.
9. Under *Scheduling*, enter a parameter for *Queue*.
10. If `job level` parameters are required:
    a. Choose the appropriate job.
    b. Fill in the desired value in *Expression*.

11. If `job chain level` parameters are required:

    a. Choose the appropriate job chain parameter under *Reference*

12. Choose *Save* to save the changes made to the job chain.

Example

Create a job chain with two steps, each step containing one job. Use a job chain parameter and map it to the parameter of the first job of step *Step 2*.

1. Choose *Definitions* > *Job Chains* in the Navigation bar.
2. Choose *New Job Chain* from the context-menu.
3. Choose the *Job Definition* tab enter *MSLN_GL_INV_IN* in the *Name* field.
4. Fill `System` into the *Queue* field.
5. Choose *MSLN_GL_INV_IN* on the *Job Chain* tab.
6. Choose *Add Step* from the context-menu.
7. Choose *Step 1*
8. Choose *Add Job* from the context-menu and choose the newly created box.
9. Enter *info* into the *Job Definition* field.
10. Choose System_Info from the list of available job definitions.
11. Fill *10* into the *Priority* field.
12. Choose *MSLN_GL_INV_IN*.
13. Choose *Add Step*.
14. Choose *Step 2*, choose *Add Job* from the context menu and then choose the newly created box.
15. Enter *Sleep* into the *Job Definition* field.
16. Choose *System_Sleep* from the list of available job definitions.
17. Enter the value *System* into the *Queue*; note that this task is optional here, as this queue has been defined as default queue of the job chain, and will be used, by default for this job as well.
18. Choose the *Parameters* tab.
19. Choose *Add*.
20. Fill *SleepTime* into the *Name* field.
21. Select *Number* in the *Type* field, the default value was *String*.
22. Fill `1` into the *Display Order* field.
23. Fill `2000` into the *Default Expression* field.
24. Choose *Save*.

25. On the *Job Chain* tab, expand *Step 2*, choose *Sleep*.
26. Expand *Time to Sleep* and select *MSLN_GL_INV_IN:SleepTime* in the *Reference* field,
27. Choose *Save & Close* to save the job chain.

Submit the job job chain

1. Navigate to *Definitions* > *Job Chains* in the Navigation bar.
2. Choose *Submit* from the context menu of the job chain 'MSLN_GL_INV_IN".
3. Fill `3000` into the *SleepTime* field; note that the default value of the job chain is `2000` and the default of the job definition is `1000`.
4. Choose *Next*, and *Next* again.
5. Fill *System* into the *Queue* field, and choose *Submit*.

104,0-1 Bot

Creating Job Chains with Parallel Jobs

Use

Parallel execution is used for independent jobs and to optimize resource usage on a system. Particularly useful for light weight jobs or for jobs that require simultaneous execution.

It is better to keep the amount of steps in a job chain to a minimum.

Procedure

1. Choose *Definitions* > *Job Chains* in the Navigation bar.
2. Choose *New* from the context menu.
3. Choose the *Job Definition* tab and fill in a name for the job chain, choose *Save*.
4. On the *Job Chain* tab, choose the name of the job chain.
5. Choose *Add Step* from the context menu, notice that a new step with the caption *Step 1* has been added.
6. Choose *Add Job* from the context menu of *Step 1*.
7. Choose *Job 1* and enter the part of the job definition name into the *Job Definition* field.
8. Choose the desired job definition out of the list of available job definitions.
9. Under *Scheduling*, enter a parameter for *Queue*.
10. Perform actions 6 to 9 of this procedure for each required job.
11. Choose *Save* to save the job chain.

Example

Create a job chain with one step containing two jobs.

1. Choose *Definitions > Job Chains* in the Navigation bar.

2. Choose *New Job Chain* from the context-menu.

3. Choose the *Job Definition* tab enter *MSLN_FC_INV_IN* in the *Name* field.

4. Choose *MSLN_FC_INV_IN* on the *Job Chain* tab.

5. Choose *Add Step* from the context-menu.

6. Choose *Step 1*

7. Choose *Add Job* from the context-menu and choose the newly created box.

8. Enter *info* into the *Job Definition* field.

9. Choose System_Info from the list of available job definitions.

10. Fill *10* into the *Priority* field.

11. Choose *MSLN_FC_INV_IN*.

12. Choose *Step 1*, choose *Add Job* from the context menu and then choose the newly created box.

13. Enter *Sleep* into the *Job Definition* field.

14. Choose *System_Sleep* from the list of available job definitions.

15. Enter the value *System* into the *Queue*; note that in this case, this task is optional. Since we did not define a default queue for the job chain, the submit wizard will require you to fill the queue field, this also means that you cannot submit this job chain automatically from an event, for example. You define a default queue for the job chain on the *Job Definition* tab, see the Creating Job Chains with Parameter Handling topic for an example.

16. Choose *Save & Close* to save the job chain.

Submit the job job chain

1. Navigate to *Definitions > Job Chains* in the Navigation bar.

2. Choose *Submit* from the context menu of the job chain 'MSLN_FC_INV_IN".

3. Choose *Next*.

4. Fill *System* into the *Queue* field, and choose *Submit*.

Creating Job Chains with Job Dependencies

Use

Complex job dependencies can be implemented with job chains and other repository objects in an intuitive way.

The following different repository objects can be used to create dependencies:

- Locks - Prevent two different jobs to run simultaneously.

- Events - Jobs can raise and clear events as well as get submitted by events.

The following job chain features create dependencies:

- Status Handlers - Job chain execution can be adapted at runtime by the use of status handlers.
- Precondition function - Allow or disallow job execution based on the output of a function.

Procedure

1. Choose *Definitions > Job Chains* in the Navigation bar.
2. Choose *New Job Chain* from the context-menu.
3. Choose the *Job Definition* tab and fill in a name for the job chain, choose *Save*.
4. On the *Job Chain* tab, choose the name of the job chain.
5. Choose *Add Step* from the context-menu, notice that a new step with the caption *Step 1* has been added.
6. Select *Step 1*.
7. Specify a precondition expression in the *Precondition* field, an empty *Precondition* field is evaluated as `true`.
8. Expand *Status Handlers* and specify an action to be performed for each desired status.
9. Choose *Add Job* from the context-menu of *Step 1*.
10. Choose *Job 1* and enter the part of the job definition name into the *Job Definition* field.
11. Choose the desired job definition out of the list of available job definitions.
12. Under *Scheduling*, enter a parameter for *Queue*.
13. Specify a precondition expression in the *Precondition*.
14. Perform actions 6 to 9 of this procedure for each required job.
15. Set dependencies:
    a. On the *Locks* tab, fill in a name of a desired lock that the job chain should hold.
    b. On the *Wait Events* tab, choose *Add* and fill the name of an event. Select a value for *Auto submit*.
    c. On the *Raise Events* tab, choose *Add* and fill the name of an event, specify a *Raiser Comment* and select a *Status To Raise On*.
16. Choose *Save* to save the job chain.

Examples

Using Locks

Two independent job chains are not to run simultaneously.

1. Creating a lock

1. Choose *Definitions > Locks* in the Navigation bar.
2. Choose *New Lock* and fill in a name.

3. Choose *Save and Close*.

2. Attaching the lock to the job chains

1. Choose *Definitions > Job Chains*.
2. From the context-menu, choose *Edit*.
3. Choose the job chain.
4. Choose the *Locks* tab, fill in the name of a previously created lock.
5. Choose *Save and Close*.

Using Events

1. Create an event
    a. Choose *Definitions > Event Definitions* in the Navigation bar.
    b. Choose *New Event Definition* and fill in a name.
    c. Choose *Save and Close*.
2. Attach an event to be raised by the job chain.
    a. Choose *Definitions > Job Chains*
    b. From the context-menu, choose *Edit*.
    c. Choose the job chain.
    d. Choose the *Raise Events* tab and choose *Add*, choose the previously created event.
    e. Choose *Save* to save the job chain.
3. Attach an event to submit the job chain.
    a. Choose *Definitions > Job Chains*
    b. From the context-menu, choose *Edit*.
    c. Choose the job chain.
    d. Choose the *Wait Events* tab and choose *Add*, choose the previously created event.
    e. Choose *Save* to save the job chain.

Using Status Handlers

A job chain with four steps should run step 4 only when one of the three previous steps failed and upon successful completion restart the job chain.

When step 4 fails an operator should be informed of the problem.

1. Choose *Definitions > Job Chains*
2. From the context-menu, choose *Edit*.

3. Under *Error*, choose *GotoStep* and step 4.
4. Edit the second step in the same way.
5. Choose the third Step, choose the *Status Handler* tab.
6. Under *Error*, choose *GotoStep* and step 4.
7. Under *Completed*, choose *MarkCompleted*.
8. Choose the last Step.
9. Under*Completed*, choose *RestartChain*.
10. Under *Error*, choose *RequestRestart*.
11. Choose *Save* to save the job chain.

Using Precondition Functions

The 1st step should run on Mondays

1. Choose *Definitions* > *Job Chains*
2. From the context-menu of the job chain, choose *Edit*.
3. Choose the first Step.
4. Fill `=Time.isTimeWindowOpen(Time.now('GMT'), 'System_Week_Monday')`
5. Choose *Save & Close* to save the job chain.

Creating Job Chains for Multiple SAP Systems

Use

The SAP jobs of a job chain require parameters, there are two parameter classes, scheduling parameters and job parameters.

Scheduling parameters define the status of the job chain at submit time, the queue and the priority of the job.

For SAP jobs, the queue defines the SAP System where the job runs, when the *SAP System* parameter is not set.

Note

You may be restricted to connecting to a single SAP System from within SAP CPS at any given time; this depends on the value of *ProcessServerService.SAP.limit* license key and your privileges. If this is the case, you will not be able to follow this procedure.

Procedure

1. Choose *Definitions* > *Job Chains* in the Navigation bar.
2. Choose *New* from the context menu.
3. Choose the *Job Definition* tab and fill in a name for the job chain, choose *Save*.

4. On the *Job Chain* tab, choose the name of the job chain.
5. Choose *Add Step* from the context menu, notice that a new step with the caption *Step 1* has been added.
6. Choose *Add Job* from the context menu of *Step 1*.
7. Choose *Job 1* and enter the part of the job definition name into the *Job Definition* field.
8. Choose the desired job definition out of the list of available job definitions.
9. Under *Scheduling*, enter a parameter for *Queue*.
10. Perform steps 6 to 9 for every job.
11. Choose *Save* to save the job chain.

Example

A job chain with two steps each containing one job will be created, the two jobs (SAP CPS job definition *SAP_Info* and SAP ABAP *RSUSR000*) will be run on separate SAP Systems (PR0 and PR1).

1. Choose *Definitions > Job Chains* in the Navigation bar.
2. Choose *New* from the context menu.
3. Choose the *Job Definition* tab, enter *MSLN_CTRL_INV* in the *Name* field.
4. Choose *New* next to the *Application* field, enter *MSLN_CTRL* into the *Name* field of the new window, choose *Save & Close*.
5. Choose the *Job Chain* tab.
6. Choose *Add Step* from the context menu of green box.
7. Choose *Add Job* from the context menu of *Step 1*.
8. Choose *Job 1* and enter *info* into the *Job Definition* field.
9. Choose *SAP_Info* from the list of available job definitions.
10. Choose the *Scheduling Parameters* tab
11. Fill `PR1_Queue` into the *Queue* field.
12. Fill `50` into the *Expression* field.
13. Choose the *Parameters* tab, and enter the parameters in the table below, only enter the values in the table, leave all other fields with their defaults.
14. Choose the *Job Chain*.
15. Choose *Add Step* from the context menu of *MSLN_CTRL_INV*.
16. Choose *Add Job* from the context menu of *Step 2*.
17. Choose *Job 1* and enter *ABAP* into the *Job Definition* field.
18. Choose *SAP_AbapRun* from the list of available job definitions.

19. Choose *SAP_AbapRun*.

20. Choose Reference *MSLN_CTRL_INV:SYSTEM* for *SAP System*.

21. Choose Reference *MSLN_CTRL_INV:ABAP* for *ABAP program name*.

22. Choose Reference *MSLN_CTRL_INV:JOBNAME* for *SAP job name*.

23. Fill `200` into the *SAP client* field. This step is optional, the default client will be used if no client is provided.

24. Choose *Save & Close* to save the job chain.

Parameter Name	**Display Order**	**Value**
SYSTEM	PR0	1
ABAP	RSUSR000	2
JOBNAME	MSLN_TEST	3

Precondition Functions

Use

Preconditions are used to determine if a step or job in a job chain is to be started or skipped.

The precondition is a boolean function (it returns "true" or "false") which is written in Redwood Expression Language.

When the precondition function returns:

- "true" or no precondition is defined, the step or job is started.
- "false", the step or job is skipped.

Note

Expressions always start with an equals sign (=).

Note

Preconditions are executed before the job is persisted, or saved in the database. This has the effect that it is not possible to access itself or, when it is a job chain, any other job in the same job chain.

Job Chains and Final Status

The final status of a job chain is determined by the status of the last step that reached a final state; the following final states are mapped to `Completed`: `Disabled`, `Skipped`, `Ignored`. You can use keywords and a registry key to toggle this behavior.

The following comments can be used to enable the new behavior:

```
@useLastExecutedStepStatus=true
```

To disable, use the following:

```
@useLastExecutedStepStatus=false
```

Just place the keyword (including @ and the desired value, true or false) into the *Comment* field on the *Comment* tab of a job chain.

The registry key that governs the default behavior:

`/configuration/jcs/jobchain/useLastExecutedStepStatus`

can be set to `true` or `false` (default).

`true` means that if you have a job chain of which the last step is `Disabled`, for example, the final status of the chain will be that of the preceding step, if this step is not in one of the previously mentioned states.

Job Chains and Restart Behavior

The Restart Behavior is only evaluated at the end of the step. When a step is skipped, due to a precondition, for example, the restart behavior is not evaluated, since the step does not run, thus does not finish. Therefore you have to put the precondition on the job, so the job goes to Skipped and not the step. This can be an issue if you use

`stepIterations < n` preconditions, where `n` is a number, in combination with restart behavior.

Procedure

1. Choose *Definitions > Job Chains*
2. From the context menu, choose *Edit*.
3. Select a step or job and choose *Edit Step* or *Edit Job*
4. Choose *Precondition* and fill in an expression
5. Choose *Save and Close*

Example

In this example, a test is done to see if the time window System_WorkDays is open at this time. The function Time.now() returns the current system date time in the specified time zone. The function Time.isTimeWindowOpen(`time`, `time_window`) tests if the time window `time_window` is open at a given time `time`.

In the below example the following is true:

- Time window `System_Week_WorkDays` is open on workdays only
- The time now `Time.now('GMT')` is 2010/08/23 10:50:44,910 GMT

If the following expression is false the job is skipped:

```
=Time.isTimeWindowOpen(Time.now('GMT'), 'System_Week_WorkDays')
```

See Also

- Redwood Expression Language
- Creating Job Chains

Redwood Expression Language

Use

Redwood Expression Language (or REL) is a simple expression language that supports basic arithmetic and comparison operators, and some predefined functions. Different scripting contexts have different predefined objects, the functions on this page are available in all contexts. Not all functions are listed here, see REL Functions for a complete list of functions.

You can create your own functions for use in REL and RedwoodScript using a library with REL entry points, see Libraries for more information.

The REL can be used to specify:

- Job definition parameter default values
- Job definition return code mappings
- Job chain step preconditions
- Job chain job preconditions
- Job chain job parameter mappings
- Job chain job scheduling parameters

Expressions in REL are designed to be very similar to expressions in RedwoodScript, and the predefined functions are the same.

Operators

Redwood Expression Language provides support for the basic arithmetic operators:

- + - addition and string concatenation
- - - subtraction
- * - multiplication
- / - division
- % - integer modulus

It also supports comparison:

- === - equal to
- !== - not equal to
- > - greater than
- >= - greater than or equal
- < - less than
- <= - less than or equal

Escape Character

- \ - escape character, and thus must be escaped if the literal is meant to be used. This means that a UNC path would be specified as follows:

  ```
  \\\\server\\share\\folder\\file
  ```

.

Predefined functions

A number of predefined functions are available, these are grouped into functional areas.

Math

- Math.abs(*x*) - take the absolute value of a number. For positive numbers, this is the number, for negative numbers this is the opposite of the number. For example: Math.abs(1) = 1, Math.abs(-10) = 10.
- Math.floor(*x*) - take the floor of a number, this is the largest integer less than or equal to x. For example: floor(2.9) = 2, floor(-2) = -2 and floor(-2.3) = -3
- Math.ceil(*x*) - take the ceiling of a number, this is the smallest integer not less than x. For example: ceiling(2.3) = 3, ceiling(2) = 2 and ceiling(-2.3) = -2.
- Math.round(*x*) - round x to the nearest integer. This uses the 'bankers rounding' algorithm to round numbers .5 exactly, such numbers are rounded to the nearest even number. For example: Math.round(2.2) = 2, Math.round(2.4) = 2, Math.round(3.5) = 4.

Range

- Range.inRange(*candidate*, *range*) *- is* candidate *in the range specified by* range". For example: Range.inRange(2, '1-3') = true, Range.inRange(4, '2-3,6-') is false.
- Range.inSet(*candidate*, *set*) *- is* candidate *in the range specified by* set". For example: Range.inSet(1, ' X X X X ') = true, Range.inSet(6, '_1_3_5_7') = false.

Ranges are made up of components, separated by commas. Each range component can be:

- x - the number *x*
- -x - all numbers from zero to *x* (including *x*).
- x-y - all numbers from *x* to *y* (including both *x* and *y*)
- y- - all numbers from *y* to the maximum value for an integer.

Example ranges:

- *1-5,7-9*
- *-5, 8-*
- *1,2,4,8,16*

Sets are made up of elements, starting at zero. If the element at position x is ' ' (space) or '_' (underscore) then that element is NOT in the set. Otherwise that element is in the set.

Example sets:

- *0 2 4 6* - contains 0, 2, 4, 6
- *X X X X* - contains 0, 2, 4, 6
- *_X_X_X_* - contains 1, 3, 5

Example expression:

- =Range.inRange(1, '0-5') is true
- =Range.inSet(2, '01_345') is false

Time

Function	**Arguments**	**Description**
Time.now()	timeZoneId	The time in the specified time zone; for example (2011/08/23 09:39:33,324 GMT)
Time.now()		The time in the specified time zone; for example (2011/08/23 09:39:33,324 GMT)
Time.isTimeWindowOpen()	DateTimeZone, timeWindowName	Is the specified time window open at the specified time.
Time.isTimeWindowOpenNow()	timeWindowName	Is the specified time window open at the current system time.
Time.expression()	DateTimeZone, expression	Apply expression to DateTimeZone
Time.expressionNow()	expression	Apply expression to Time.now()

Time expressions are a sequence of one or more operations, applied in order to a time (generally now, but may be a specified date).

The following time expressions are available:

- set <specifier> <value> - set <specifier> to <value>
- add <value> <specifier> - add <value> to <specifier> (This may propagate)
- subtract <value> <specifier> - subtract <value> from <specifier> (This may propagate)
- truncate <specifier> - truncate at <specifier> (This will zero <specifier> and everything below it).

The <value> is always a number:

- The days of the week start at 1 for Sunday.
- The days of the year start at 1 for 1 January.

The <specifier> is one of:

- add/subtract: second, minute, hour, day, week, month, year
- truncate: second, minute, hour, day, month
- set: second, minute, hour, day, dow, day_of_year, week, week_of_month, month

The *day* is always the day of the month, and *dow* is the day of the week.

Note

The date you enter is treated in a lenient fashion, that is `32/12/2011` is treated as `01/01/2012`.

Example time expressions:

- add 1 minute
- add 3 seconds
- set hour 1
- truncate day
- subtract 2 days

Example using the Time functions

In these examples, the current time is 2011/08/23 10:50:44,910 GMT according to output of the `=Time.now('GMT')` function.

Example 1

```
=Time.isTimeWindowOpen(Time.now('GMT'), 'System_Week_WorkDays')
```

This expression returns true if at evaluation-time it is a workday in the GMT time zone. An expression like this might be used as a job chain step (or job) precondition to prevent certain steps from running on weekends.

Example 2

```
=Time.expressionNow('add 2 hours add 1 minute add 1 second subtract 2 days')
```

This expression added 2 hours, 1 minute a second and subtracted 2 days from the current time and returned `2011/08/21 13:51:45,910 Europe/Paris`

Example 3

```
=Time.expression(Time.now('GMT'), 'truncate minute')
```

This expression returned: `2011/08/23 10:50:00,000 GMT`

Example 4

```
=Time.expression(Time.now('GMT'), 'set minute 13')
```

This expression returned: `2011/08/23 10:13:44,910 GMT`

Example 4

```
=Time.expression(Time.now('GMT'), 'subtract 1 month')
```

This expression returned: `2011/07/25 10:13:44,910 GMT`

Example using the Time functions as a precondition:

```
=Time.isTimeWindowOpen(Time.now('GMT'), 'System_Week_WorkDays')
```

The function returned *false*, so the job is skipped

Strings

String functions are all applied to a string instance. Strings are treated as arrays of characters, with the first index at 0 and the final index at length-1.

- aString.concat(*s1*, [*s2*[, ...]]) - concatenate *s1*, *s2*, ... to *aString*. This function takes one or more arguments. For example: 'a'.concat('b') = 'ab', 'a'.concat('b', 'c', 'd') = 'abcd'
- aString.charAt(*pos*) - return the character at *pos* in *aString*. The index *pos* is zero based. For example: '012345'.charAt(2) = '2'.
- aString.indexOf(*candidate*[, *startIndex*]) - return the first index of *candidate* in *aString*. If *startIndex* is not specified, the search starts at 0, otherwise it starts at *startIndex*. For example: '012345345'.indexOf('3') = 3
- aString.lastIndexOf(*candidate*[, *startIndex*]) - return the last index of *candidate* in *aString*. If *startIndex* is not specified, the search starts at 0, otherwise it starts at *startIndex*. '012345345'.lastIndexOf('3') = 6
- aString.split(*separator*, *limit*) - slice *aString* using separator, returning at most *limit* entries. For example: '1,2,3,456,7'.split(',', 4) = ['1', '2', '3', '456,7']
- aString.substring(*start*[, *end*]) - return a substring of *aString* starting at *start*. If *end* is not specified the remainder of the string is returned, otherwise it is the last index to return. '01234'.substring(2,4) = '234'
- aString.toLowerCase() - convert the string to lower case. For example: 'AbC'.toLowerCase() = 'abc'
- aString.toUpperCase() - convert the string to upper case. For example: 'aBc'.toUpperCase() = 'ABC'

Note

Like in Java, the + (plus) character can be used for concatenation of strings. For example `'Hello '+'Universe!'`

Casts

Change a data type to another.

- ToBoolean(*x*) - convert *x* to a boolean.
- ToNumber(*x*) - convert *x* to a number.
- ToInteger(*x*) - convert *x* to an integer.
- ToString(*x*) - convert *x* to a string.

Note

Expressions always start with an equals sign (=).

Constraints

The following functions are available for validating parameter values. You specify these functions in the *Simple Constraint Data* field of a parameter; note that the *Simple Constraint Type* must be set to *Expression* for these functions to work.

Constraint.listConstraint(titles, list, bool)

Constraint.pairListConstraint(titles, list, bool)

- titles is a comma separated list of titles.
- list is either a normal list or a pair list
- bool is a boolean expression involving the parameter value

```
=Constraint.listConstraint('Country', 'de,nl,gt', value !== '')
=Constraint.pairListConstraint('Country, Tel. Code', 'de=49,nl=31,gt=502', value !== '')
```

Tables

The following functions are declared within the `Table` object and allow you to interact with tables, such as the *System_Variables* table.

- getColumnString(table, key, column)
- getRow(table, key)
- getColumnNumber(table, key, column)
- formatRow(table, key, rowStart, columnFormat, columnSeparator, rowEnd)
  - *columnFormat* - accepts strings as well as the following substitution parameters to specify the position of the elements:
    - *{0}* - column header
    - *{1}* - value
    - *{2}* - columnSeparator (should always be the first)

The following examples use the *Countries* table

Countries Table:

Key	Name	Abbreviation	Translation	Conversion_Rate
31	The Netherlands	NL	Nederland	1.0
33	France	Fr	France	1.0
49	Germany	De	Deutschland	1.0
502	Guatemala	Gt	Guatemala	11.6

Note

The *Key* column is a special built-in column; it is used by the table functions, for example, to look up data in the table and does not need to be specified in the table definition.

getColumnString

```
=Table.getColumnString('Countries', '31', 'Translation')

Returns:
Nederland
```

getColumnNumber

```
=Table.getColumnNumber('Countries', '502', 'Conversion_Rate')

Returns:
11.6
```

getRow

```
=Table.getRow('Countries', '33')
Returns:
FrFranceFrance
```

formatRow

```
=Table.formatRow('Countries', '49', '#', '{2}{0}={1}', '|', '#')

Returns:
#Abbreviation=De|Conversion_Rate=1.0|Name=Germany|Translation=Deutschland#
```

```
=Table.formatRow('Countries', '49', '||', '{2}{1}', '#', '||')

Returns:
||De#1.0#Germany#Deutschland||
```

```
=Table.formatRow('Countries', '49', '', '{2}{1}', '\n', '')

Returns:
De
1.0
Germany
Deutschland
```

The following Redwood Expression Language function allows you to retrieve values out of the built-in *System_Variables* table:

- Variable.getString(key)

It is a shorthand for `Table.getColumnString('System_Variables', key, 'SystemValue')`

Queries

Availability: Requires the Chargeable version

The following functions allow you to query the database, these functions do not return iterators, only the first value of the first column.

- Query.getString(query string, [bind variables])
- Query.getNumber(query string, [bind variables])

```
=Query.getString('select Job.Description from Job where Job.Description = ?', ['GL_REP_EMEA'])
```

```
=Query.getNumber("select Job.JobId from Job where Job.Description = GL_REP_EMEA')
```

The following function returns a job object, it is to be used in a job chain.

Job chain job names take the form Step name, job job number. For example: "Step 2, job 3" or "Data load, job 1".

Query.getRelativeJob(jobname string)

```
Job jcJob = Query.getRelativeJob("Data load, job 2");
```

JobChainParameters

- chainParameters.<parameter_name>
- chainOutParameters.<parameter_name>
- chainJobId
- JobChainParameters.getOutValueNumber(String jobName, String parameterName)
- JobChainParameters.getOutValueString(String jobName, String parameterName)
- JobChainParameters.getOutValueDate(String jobName, String parameterName)

JobChainParameters.getOutValueString('backup, Job 1', 'connectString')

Job chain job names take the form Step name, job job number. For example: 'Step 2, job 3' or 'Data load, job 1'.

Get the value of the Date output parameter bckDate of the job named *Backup, Job 1* in the current job chain, which is, by default, the first job of the step named *Backup*.

Availability: Requires the Chargeable version

```
JobChainParameters.getOutValueDate('Backup, Job 1', 'bckDate')
```

Get the value of the Number output parameter BckSize of the job named *Backup, Job 1* in the current job chain, which is the first job of the step called *Backup*.

Availability: Requires the Chargeable version

```
JobChainParameters.getOutValueNumber('Backup, Job 1', 'bckSize')
```

Get the value of the String output parameter connectString of the job named *Backup, Job 1* in the current job chain, which is the first job of the step called *Backup*.

Availability: Requires the Chargeable version

```
JobChainParameters.getOutValueString('backup, Job 1', 'connectString')
```

See Also

- Scripting contexts
- RedwoodScript

Submitting Jobs and Job Chains

Use

When a user requests a job definition to be executed, a process server will have to schedule the job for execution. A request for execution is also called "submitting". Before the execution request is passed on to the process server, the user may:

- Change scheduling behavior parameter values.
- Enter the job parameter values.

Once a request has been submitted, the job can be created by process server. By default, a scheduled job will be executed according to the start date and time, and provided a scheduler is up and running. Jobs can also be scheduled directly on hold, this job will require human intervention before it will start.

See Also

1. Submitting a Job Definition or Job Chain

Submitting a Job or Job Chain

Use

When you request a job definition to be executed, a process server schedules the job for execution. A request for execution is also called "submitting". Before the execution request is passed on to the process server, you may:

- Change scheduling behavior parameter values.
- Enter the job parameter values.

Once a request has been submitted, the job can be created by the process server. By default, a scheduled job is executed according to the start date and time, and provided a scheduler is up and running. Jobs can also be scheduled directly on hold, this job requires human intervention before it starts.

To submit a job with a schedule, to repeat periodically, you can use the built-in frequencies, *daily*, *weekly*, *monthly* or *yearly*. If you want the job to repeat multiple times a day, for example every two hours, you must first create a submit frame.

Procedure

There are several ways you can submit a job:

1. Choose the *Submit* button,

   a. on the tool-bar to submit any job definition. A selection screen appears where the job definition that is to be submitted must be selected.

   b. in the lower detail pane of *Definitions* > *Job Chains* or *Definitions* > *Job Definitions* to submit the selected job definition or job chain.

   c. from the context menu of a job, job definition or job chain.

2. Choose the *Submit* button to submit the job immediately.

3. Choose the navigation links, if parameters or the scheduling behavior are to be changed.

Note

Built-in SAPR3 job definitions cannot be submitted directly, as they do not have a default queue. The queue of built-in SAPR3 job definitions defines the SAP System where the job must run, therefore the queue is only set by default when there is only one SAP System defined.

Submit a job every 5 minutes

1. Navigate to *Environment* > *Submit Frames*.
2. Locate a submit frame that opens every 5 minutes, if there is none, you will have to create one.
   a. Choose *New submit frame* from the context menu.
   b. Fill a name into the *Name* field, SAP recommends that in this case you use the name *Every5Minutes*.
   c. On the *Elements* tab, choose *Interval*, fill 5 into the *Submit every* field and select *Minutes*.
   d. Choose *Save & Close*.
3. Navigate to *Definitions* > *Job Definitions* and choose *Submit* from the context menu of the job definition you want to schedule.
4. Fill in any parameters, if required, and choose *Next*.
5. On the *Times and Dates* page, select *Submit frame* and fill in the name of the submit frame, in this case *Every5Minutes*.
6. Choose *Submit*, you may have to select a queue and choose *Submit* again, depending on the job definition you selected.

Parameters

When the job definition has no parameters, the *Parameters* navigation link is disabled and appears grayed out.

- If the parameters are assigned to groups, the group names are shown on the tabs, located at the top of the window.
- If the parameters have default values, they are set initially.
- Enter the values for the parameters or choose them from the drop-down lists.
- Alternatively, you may use the *Default* or *Last used values* as follows:
  - *Last used values* - Sets the job parameter value to the values of the last job of this job definition that you submitted.
  - *Default values* - Sets the job parameter value to the default expression value of the corresponding job definition parameter. If there is no default expression filled out, the parameter will be cleared.

Time and dates

Jobs are submitted with a start time calculated relative to the session time zone. The time zone you choose at submit-time will become the job time zone.

To schedule the job, select the date, time and time zone, by default this is the time the job is created in the database and displayed as *Now*. A *Calendar* is available to choose a date.

Note

The date you enter is treated in a lenient fashion, that is `32/12/2011` is treated as `01/01/2012`.

In order to schedule multiple executions of a job definition, there are several options:

- *Once* (default)
- *Daily*
- *Weekly*
- *Monthly*
- *Yearly*
- *Dates*
- *Advanced*

Daily, Weekly, Monthly, Yearly

- Select the interval.

Select the range.

- *No end date* means the job is rescheduled endlessly.
- *End after* means the job is rescheduled a fixed number of times.
- *End by* means that the job is rescheduled until a fixed date.

Select the scope

- Scope - *first, second, third, fourth or last*
- Day, Week, Month
- Interval (of every)
  - Skip a number of months

Dates

Choose *Dates* if specific dates are to be used for when the job is submitted.

Advanced

Choose *Advanced* if predefined time window and submit frame are to be used for the job.

Scheduling

The following Scheduling Parameters are available:

- *Queue* - The queue in which the job is executed.

- *Priority* - A job can be assigned a particular priority between 0 and 100, where 0 is the minimum priority and 100 the maximum priority. When system load becomes too high, the process server can postpone low-priority jobs and continue executing high-priority jobs.
- *Hold On Submit* - To submit the job on status *Hold*, manual intervention is necessary to start the job.

Submit

The following steps are available after submitting the job:

- *I am done* - The job is submitted and the window is closed.
- *Show all jobs* - The job is submitted, the window is closed and the Job monitor is shown. Choose **Last** to see the job in the overview.
- *Submit another job* - The job is submitted, the window remains open, and another job definition can be selected again.

See Also

- Creating Job Chains
- Using Recurrences
- Using Submit Frames in Jobs
- Defining Execution Times
- Timezones
- Monitoring Automated Processes

Scheduling with Times and Dates

Job execution time can be controlled in SAP CPS by the use of the following:

- Submit frames - Submit frames define when and how often a job should be automatically resubmitted.
- Time windows - Time windows define in which period of time a job is allowed to start.
- Time zones - Time zones help define the date and time accurately in large environments with clients and/or process servers split across multiple geographical time zones.

Job definitions, job chains and jobs can have a submit frame, a time window and a time zone. Submit frames can also have a time zone and a time window, time windows can have time windows and a time zone. Complex scheduling can easily be achieved by the use of a combination of submit frames and time windows, the time zone further defines the time window.

See Also

1. Rescheduling Jobs Automatically
2. Defining Execution Times
3. Time zones

Rescheduling Jobs Automatically

Use

There are two ways to have jobs resubmit automatically, recurrences and submit frames. Recurrences have the advantage to be very easy to configure but the disadvantage to create a job immediately for every occurrence for the next two years, if applicable. Submit frames are separate objects that need to be created but that can be reused by many jobs. Jobs are resubmitted as soon as the job has finished.

See Also

1. Using Recurrences
2. Creating Submit Frames
3. Using Submit Frames in Jobs

Using Recurrences

Use

Recurrences are used to automatically reschedule jobs in a trivial way. Redwood Software recommends the use of submit frames instead, as they offer greatly flexibility. The other main disadvantage in using recurrences is that all future jobs for the next two years are immediately be displayed in the job monitor. When submit frames are used, only the following job is visible in the job monitor.

Special care must be taken when editing a job that is part of a recurrence, as changes you make might get applied to all non-final jobs of the occurrence.

Changes made to any of the following parameters will be applied to all non-final jobs of the recurrence:

- Job.ForcedProcessServer
- Job.Priority
- Job.Queue
- Job.TimeWindow
- Job.TimeWindowTimeZone
- JobParameter.InValue

When you decrease the number of recurrences, any surplus jobs will be canceled.

Procedure

1. Choose *Definitions* > *Job Definitions*.
2. From the context menu of a job definition, choose *Submit*.
3. Choose *Time and Dates*.
4. Choose *Daily*.
5. Fill in the fields.

6. Choose *Submit*

Example

1. Choose *Definitions > Job Definitions*.
2. From the context menu of the *System_info* job definition, choose *Submit*.
3. Choose *Time and Dates*.
4. Choose *Daily* to end after 10 occurrences.
5. Choose *Submit*

See Also

- Creating Submit Frames
- Using Submit Frames in Jobs

Creating Submit Frames

Use

A submit frame is a set of rules that control the repetitive execution of a job. When a job definition is submitted and a submit frame is used, the corresponding jobs are restarted automatically by a process server according to the frequency defined in the submit frame.

Simple multiple executions of a job definition can also be achieved by specifying a recurrence. A job recurrence is specified for one or more jobs when you submit a job definition. A submit frame is an object that can be reused for multiple jobs.

Submit frames are part of the 'Environment' section in the navigation bar. The following standard submit frames are provided with SAP CPS:

- *System_DeleteJobFilesSubmitFrame* - Submit frame for the system job that deletes job files
- "System_Get_Tutorials_listSubmitFrame - Submit frame for system job that fetches the tutorials list
- *System_Ignored_Alert_ReporterSubmitFrame* - Submit frame for Ignored_Alert_Reporter system job
- *System_ProcessKeepClausesSubmitFrame* - Submit frame for the system job that processes keep clauses
- *System_SynchronizeBackgroundJobsSubmitFrame* -Submit frame the system job that synchronizes background jobs
- *System_UpdateJobStatisticsSubmitFrame* - Submit frame for the system job that updates job statistics

Submit Frame Elements

A submit frame is made of one or more submit frame elements defining the cycle. If more than one submit frame element is specified, the total cycle is a union of all submit frame elements. If submit frame elements overlap, only one job is created.

This means that if a submit frame has two elements, *every hour* and *every 3 days*, for example, the job will run every hour of every day. In this example, every time a job in the job group has reached a final status (which does not break the submit frame according to *Restart Behavior*), SAP CPS evaluates the submit frame based on the first job of the job group, checking the union of all its elements and determining the next date/time it can submit a new job for. The equation is as follows:

```
<submit_base_time> + (<n> x <amount> <units>)
```

- <submit_base_time> - the original date/time the first job of the job group was started
- <n> - the number of cycles required to get a date/time after the time now
- <amount> - the you entered in the *every* field in the submit frame element
- <unit> - the time unit you chose in the *Unit* field (seconds, minutes, hours ...)

The *every 3 days* element is irrelevant, in this case; **no** additional jobs will be created based on this element because it overlaps every time with the other. Note also, that if the job takes longer than `<amount> <unit>`, in this case 1 hour, or for another reason goes past the start time of the following job, the job will be skipped.

Another example: a submit frame has two elements, *every hour during time window System_Week_WorkDay* and *every 3 days*. With this submit frame, jobs will run every hour during *System_Week_WorkDay* and every time the following equation results in a day that is not a *System_Week_WorkDay*, the job will run on that day:

```
<day_of_original_request> + (<n> x 3 days)
```

In fact, in this example it means every third Sunday and Saturday.

Note

Changes to the submit frame will affect the current job if it has not yet started, otherwise it will be the next.

See the Frequently Asked Questions topic for more information.

Actions

Submit frames support the following actions:

Action	Description
Export	Export the submit frame into a CAR file
Edit	Edit the submit frame
Edit Security	Edit the security of the submit frame
Delete	Delete the submit frame
Duplicate	Make a copy of the submit frame to create a similar one
Expand All	Expand all submit frames in the current filter
New	Create a new submit frame
Filter > New Filter	Create a new submit frame filter
Filter > Edit Filter	Edit current submit frame filter
Filter > Delete	Delete current submit frame filter

Finding Submit Frames

You can search for submit frames using filters and the *Search Submit Frames* box on the Submit Frames tab. This box is known as the *intelliSearch* box and located under your username on the top right-hand side of the user interface. Filters allow you to specify a list of objects with static criteria. IntelliSearch allows you to specify complex queries in a simple way using prefixes. Prefixes are used to specify which property you are searching in and have short and long syntaxes. For example, if you want to display all submit frames with the term *import* in the comment, you would use the search criteria as follows:

```
c:import
```

You can search more than one property, as follows:

```
c:import n:Bi
```

Note

No spaces should be entered before or after the colon (*:*).

See the Advanced Object Search for more information.

The following table illustrates the available prefixes for submit frames:

Prefixes	**Description**
n, name	searches the name property
c, com, comment	searches the comment property
d, desc, description	searches the description property

Deleting Submit Frames

You can only delete submit frames when no other objects relate to them. For example, if there are jobs that have run with the submit frame, the submit frame cannot be deleted until all jobs that ran with it have been deleted. You can see all jobs that relate to the submit frame in *Related Objects* in the lower detail pane and on the show page.

The table in related objects contains three columns:

- Type - the type of object with a link to it
- Related Object - the name of the object with a link to it
- Used As - objects can sometimes be used in different roles

Security

Privilege	**Description**
SubmitFrame.Create	Create submit frames
SubmitFrame.Delete	Delete submit frames
SubmitFrame.Edit	Edit submit frames
SubmitFrame.View	Access submit frames

You can grant privileges on two levels, *Access* and *Admin;* a privilege granted on *Admin* level allows the grantee to grant the privilege to other users. These privileges can be granted *system-wide*, per *partition* or *isolation group*.

If you have the security module, which requires the *Module.Security* license key, you have an additional *Security* tab on the submit frame. It allows you to specify which users can access, edit, and delete the submit frame.

- Granting and Revoking System Privileges

- Granting or Revoking Object Privileges

Procedure

1. Choose *Environment > Submit Frames*.
2. Choose *Refresh* to see all existing submit frames.
3. Choose *New* to create a submit frame.
4. Fill in the details, refer to the section below.
5. Choose *Save and Close* to save the submit frame.

Submit Frame

- *Name* - The name of the submit frame.
- *Application* - The name of the application used to group this object.
- *Description* - The description of the submit frame.
- *Comment* - A comment about the submit frame.

Elements

- *Every Number Of* - The cycle interval in seconds, minutes, days or months. The unit of measure is specified in the Unit field.
- *Unit* - The unit of measure: seconds, minutes, days or months.
- *During Time Window* - An optional time window during which the submit frame interval can generate submits.
- *Skip when time window is closed* - If this option is selected, the submit frame element must fall inside the defined time window. If it is not selected, the first submit frame element occurs as soon as the time window is opened. This option is only applicable if *During Time Window* is defined in the submit frame element or a time window is defined on the job definition.
- *Start Time* - The start time of the job, this is only available when the *Unit* is set to *Days, Months or Years*.

Note

Skip when time window is closed has an influence on the job when the latter has a time window, defined in the submit frame element or on the job definition.

Example

1. Choose *Environment > Submit Frames* in the Navigation bar.
2. Choose the *New* button and fill in the values below for the *Submit Frame* tab.
3. Choose the *New* button and fill in the values for the *Elements* tab.
4. Choose *Save & Close* to save the new submit frame and close the edit window.

Submit Frame tab

```
Name: EveryHour
```

```
Description: Every hour
```

Elements tab

```
Every number of: 1
Unit: Hours
Must Fit Time Window: Unchecked
```

See Also

- Scheduling
- Defining Execution Times with Time Windows
- Using Recurrences

Using Submit Frames in Jobs

Use

Submit frames are used to automatically resubmit jobs. There are several ways to attach submit frames to jobs:

- Edit the job definition or job chain
- Define the submit frame at submit time
- Edit the job in the job monitor

Note

To edit a job, the job must not have reached he *Queued* status, *Scheduled* for example.

Procedure

To define a submit frame for a job chain:

1. Choose *Definitions* > *Job Chains*.
2. From the context menu of a job chain, choose *Edit*.
3. Choose *Edit Job Chain*.
4. Fill in the *Submit Frame* field.
5. Choose *Save & Close*

To define the submit frame at submit time:

1. Choose *Definitions* > *Job Chains*.
2. From the context menu of a job chain, choose *Submit*.
3. Fill in appropriate values for parameters if applicable and choose *Time and Dates*.
4. Choose *advanced* and enter a submit frame.

5. Choose *Scheduling*.

6. Choose an appropriate queue.

7. Choose *Submit*.

To edit the job in the job monitor

1. Choose *Monitoring* > *Jobs*.

2. From the context menu of a job chain, choose *Edit*.

3. Choose *Time and Dates*.

4. Choose *advanced* and enter a submit frame.

5. Choose *Submit*.

Frequently Asked Questions about Submit Frames

Use

1) Consider the following submit frame:

Element: every minute (no time window)

Element: every hour (no time window)

Q) What happens? Is a job using this running every minute or every hour? Does the order matter?

A) the job would run every minute the job would be submitted for the first occurrence from the set of [1 minute or 1 hour]. As the set returned by [1 minute] is a true superset of the set returned by [1 hour], the 1 hour element can be ignored.

2) Consider the following submit frame:

Element: every minute (no time window)

Element: every hour during time window System_Week_Friday

Q) What happens on Friday? Is a job using this running every minute or every hour on Friday?

A) the job would run every minute again the [1 minute] set also contains every occurrence of [1 hour on Fridays]. The 1 hour on Fridays element can be ignored.

3) Consider the following submit frame:

Element: every minute (no time window)

Element: every hour during time window System_Week_Friday

Element: every 30 minutes (no time window)

Q) What happens on Tuesday? Is a job using this running every minute or every 30 minutes?

A) The job would run every minute the [1 minute] set contains ever occurrence of [1 hour on Fridays] and [30 minutes]. Both these elements can be ignored

4) Consider the following submit frame:

Element: every minute during time window System_Week_Friday

Element: every hour during time window System_Week_WorkDays

Element: every 30 minutes (no time window)

Q) What happens on Friday? Is a job using this running every minute, every hour or every 30 minutes?

Q) What happens on Tuesday? Is a job using this running every hour or every 30 minutes?

A) On Fridays the job would run every minute. The every hour element is made obsolete by the 30 minute element. But on Fridays, the job would run every minute.

Defining Execution Times with Time Windows

Use

A time window defines a time interval in which something can occur: for example the time a specific job definition is allowed to run, the opening time of a queue. A time window is made up of time window elements. Each of these elements defines a period of time. These elements can used fixed dates and times, rules such as "the first working day of the month", or a combination of both (dates and times and rules). A set of time window elements does not need be continuous.

The "enabled during" time window restricts the open period of time window to the open period of the "enabled during" time window. The "disabled during" time window restricts the open period of time window to the period where the "disabled during" time window is closed. This holds for both time window and time window element.

Time windows are part of the 'Environment' section in the navigation bar. A number of standard time windows are provided with SAP CPS representing common holiday calendars and time periods. The prefixes used for these are:

- *System_BE* - Belgium specific holidays and fiscal periods.
- *System_CA* - Canada specific holidays and fiscal periods.
- *System_DE* - Germany (and German state) specific holidays and fiscal periods.
- *System_US* - United States of America specific holidays and fiscal periods.
- *System_GB* - Great Britain specific holidays and fiscal periods.
- *System_NL* - Netherlands specific holidays and fiscal periods.
- *System_Christian* - Christian holidays used in some countries.
- *System_World* - World holidays used in some countries.
- *System_* - other system time windows.

Note

Nesting of time windows should be kept to a minimum. When deep nesting is used, the processing time for the job can increase significantly.

Note

If the specified "day from" is after the last day of the month, it defaults to the last day of the month. This can have consequences for months with only 30 days and especially for February.

Note

When you define a time window that is valid in a single time zone, it is recommended to set that time zone in the time window.

Actions

Time windows support the following actions:

Action	Description
Export	Export the time window into a CAR file
Edit	Edit the time window
Edit Security	Edit the security of the time window
Delete	Delete the time window
Duplicate	Make a copy of the time window to create a similar one
Expand All	Expand all time windows in the current filter
New	Create a new time window
Filter > New Filter	Create a new time window filter
Filter > Edit Filter	Edit current time window filter
Filter > Delete	Delete current time window filter

Time Window Preview

In the preview pane of a time window, a calendar displays the current month and the days and time the currently selected time window is open. It shows the time of the current day by default but different days, months, and years can be selected. It shows a blue bar when a time window is open, a light blue bar when a time window opens and a white bar when the time window is closed. This can help fine tune a time window and can help you make sure you have made no mistakes when you created the time window.

The time scale below the calendar shows the time of the currently selected day the time window is open,

An overview table is displayed on the right-hand side and gives you an overview over all the times and dates of the current month the time window opens and closes.

When you use restrictions, for example *from Monday to Tuesday*, and the *from* value comes after the *to* value, the process server will automatically wrap. The following list illustrates the behavior:

- A range of months wraps around a year, for example, *from November to February* will be interpreted as *from November (this year) to February (next year)*
- A range of days of the month (with an offset) wraps around a month, for example "from 26

(this month) to 5 (next month)"

- A range of weekdays (with an offset) wraps until the following week.
- A range of times of day (with an offset allowing to cross midnight) *from 23:00 (today) to 2:00 (tomorrow)*

Note

Offsets are specified in hours and wrap at 24.

Finding Time Windows

You can search for time windows using filters and the *Search Time Windows* box on the Time Windows tab. This box is known as the *intelliSearch* box located under your username on the top right-hand side of the user interface. Filters allow you to specify a list of objects with static criteria. IntelliSearch allows you to specify complex queries in a simple way using prefixes. Prefixes are used to specify which property you are searching in and have short and long syntaxes. For example, if you want to display all time windows with the term *import* in the comment, you would use the search criteria as follows:

```
c:import
```

You can search more than one property, as follows:

```
c:import n:Bi
```

Note

No spaces should be entered before or after the colon (*:*).

See the Advanced Object Search for more information.

The following table illustrates the available prefixes for time windows:

Prefixes	**Description**
n, name	searches the name property
c, com, comment	searches the comment property
d, desc, description	searches the description property
a, app, application	searches the application property

Deleting Time Windows

You can only delete time windows when no other objects relate to them. For example, if there are jobs that have run with the time window, the time window cannot be deleted until all jobs that ran with it have been deleted. You can see all jobs that relate to the time window in *Related Objects* in the lower detail pane and on the show page.

The table in related objects contains three columns:

- Type - the type of object with a link to it
- Related Object - the name of the object with a link to it
- Used As - objects can sometimes be used in different roles

Security

Privilege	**Description**
TimeWindow.Create	Create time windows

TimeWindow.Delete	Delete time windows
TimeWindow.Edit	Edit time windows
TimeWindow.View	Access time windows

You can grant privileges on two levels, *Access* and *Admin;* a privilege granted on *Admin* level allows the grantee to grant the privilege to other users. These privileges can be granted *system-wide*, per *partition* or *isolation group*.

If you have the security module, which requires the *Module.Security* license key, you have an additional *Security* tab on the time window. It allows you to specify which users can access, edit, and delete the time window.

- Granting and Revoking System Privileges
- Granting or Revoking Object Privileges

Procedure

1. Choose *Environment* > *Time Windows*.
2. Choose *Refresh* to see all existing time windows.
3. Choose *New* to create a time window.
4. Fill in the details, refer to the section below.
5. Choose *Save and Close* to save the time window.

Values

The configuration details for time windows can be found in the Configuration Data for Time Windows section of the documentation.

Example

The BRLN_AIXS4 is going to be upgraded and a time window needs to be created.

1. Choose *Environment* > *Time Windows* in the Navigation bar.
2. Choose the *New* button and fill in the values for the *Time Window* tab.
3. Choose the *Elements* tab and choose *New* and fill in the values for the *Elements* tab.
4. Choose *Save and Close*.

Note

You can choose *Calendar* to choose a *from* and *until* day.

Time Window tab

```
Name: DataWareHouseUpgrade
Description: Data Warehouse Upgrade
Comment: BRLN_AIXS4 will be upgraded and unavailable the 15th of march
```

Elements tab

```
from: 15-03-2010 17:30
until: 15-03-2010 20:30
```

Tip

SAP CPS contains many predefined time windows that can be used to create new time windows. However, never remove built-in time windows.

You want to have a job run on every third workday of the month according to an SAP Factory Calendar

1. Choose *Environment > Time Windows*.
2. Choose *Refresh* to see all existing time windows.
3. Choose *New* from the context menu to create a time window, fill `PR1_Never` into the *Name* field and check the field *Is Calendar*.
4. Choose *Save & Close* and choose *New* from the context menu.
5. Fill `PR1_FR` into the name field and check the field *Is Calendar*.
6. On the *Elements* tab, fill `SAP_FactoryCalendar` into the *and period function* field, fill `calendar=PR5 FR` into the *with parameters* field.
7. Choose *Save & Close* and choose *New* from the context menu.
8. Fill `PR1_3RD_Workday` into the *Name* field, check the *Is Calendar* field.
9. Fill *PR1_Never* into the *If an open time window is closed during* field.
10. Choose *Forward* in the *then shift this day* field, fill `3` and `PR1_FR` into the *to the* and *open day in* fields.
11. Choose *Save*, inspect the open times on the *Opens at* tab to make sure they are ok.
12. Choose *Save & Close*.

Configuration Data for Time Windows

Use

The following sections describe the available settings for time windows.

Time Window

- *Name* - The name of the time window.
- *Application* - The name of the application used to group this object.
- *Description* - The description of the time window.
- *Comment* - A comment about the time window.
- *Time Zone* - The time zone for the time window.
- *Enabled During Time Window* - The time window during which this time window is enabled.

- *Disabled During Time Window* - The time window during which this time window is disabled.
- *isCalendar* - Shifting is used in the time widow.
- *If an open day is closed during* - The time window used to compare with the calendar time window. (Needs to be a Calendar time window)
- *then shift this day* - The direction of the shift.
- *to the* - The number of units to shift (has to be a positive integer)
- *open day in* - The time window that defines the units to shift.(Needs to be a Calendar time window)

Elements

Each section of the time window dialog box restricts the time the time window is open. The time window is only open when all the restrictions allow it to be open. For example, a time window that is open from 1 July to 1 August in the first section, and from the 1st Monday to the 1st Monday, then the window is open only for the first Monday in July.

Each restriction requires a 'from' and an 'until'. If these are set to the same value then the range is open for that time. In the previous example the weekdays were set as 'from the 1st Monday to the 1st Monday'.

The restrictions can be:

- A range of dates
- A range of months
- A range of days of the month (with an offset)
- A range of weekdays (with an offset)
- A range of times of day (with an offset allowing to cross midnight)

Offsets are specified in hours,wrap at 24, and allow you to shift the start of the day from midnight

to some other time during the day; for example 6:00AM or 8:00AM. Negative offsets or offsets with a value higher than 24 are not allowed.

It is also possible to specify a period function. A period function is a function defined by Redwood that allows more complicated date cycles (Easter, for example) to be calculated. Period functions can also take parameters, the interpretation of which is specific to the period function.

Time windows have an optional time zone. If this is specified then all references to the time window will use this time zone. If a time zone is not specified then a contextual time zone is used (for example the time zone of the job or queue that the time window is attached to).

Time windows can also be combined using inclusion and exclusion:

- *Enabled During Time Window* - The time window during which this time window element is enabled.
- *Disabled During Time Window* - The time window during which this time window element is disabled.

Using Time Windows in Jobs

Use

Jobs and job chains can have time windows to restrict the start time to specific time periods. Time windows restrict the start time of a job, a time window open from 10:00 until 10:05 forces the scheduled job to start at 10:00 if resources are available but the job must have started by 10:05 or the job gets the Overdue status.

Job definitions and job chains can be configured to run according to a specific time window, that can be overruled at submit time.

Note

When using a time window that does not have a time zone, it is recommended to define a time zone as well.

Procedure

To restrict a job chain to a specific time window:

1. Choose *Definitions > Job Chains*.
2. From the context menu of a job chain, choose *Edit*.
3. Choose *Edit Job Chain*.
4. Enter a time window into the field *Restrict to Time Window* of the *Job Chain* tab.
5. Choose *Save & Close*.
6. Choose *Apply*.

To overrule the time window at submit time:

1. Choose *Definitions > Job Chains*.
2. From the context menu of a job chain, choose *Submit*.
3. If the job chain has parameters, fill these in and choose *Time and Dates*.
4. Choose *advanced*, notice that the *Time Window* field is filled with a time window.
5. Enter a new time window.
6. Choose *Scheduling* and enter a queue name.
7. Choose *Submit*.

To verify which time window a specific job is restricted to:

1. Choose *Monitoring > Jobs*.
2. Select the job.
3. If the job is part of a job chain, in the lower detail pane, choose the *job id*.
4. The time window is displayed in the lower detail pane.

Defining Advanced Execution Times with Period Functions

Use

SAP CPS features rich calendaring possibilities; time windows offer great flexibility while remaining easy to define. However, sometimes complex calculations are required to determine if certain conditions are met for a time window to be open, Easter for example. For this matter it is possible to create period functions. Period functions are boolean functions used in time windows that calculate when a time window is supposed to be open or closed. Period functions are complex functions written in RedwoodScript, stored in the Repository, that are used in time windows to calculate the dates and times that a time interval is open.

When you import SAP Calendars (like the Factory and Holiday calendars), these become available as period functions.

The following period functions are built-in:

Period Function	Description
SAP_FactoryCalendar	Use the SAP calendar from an SAP instance to schedule jobs; the calendaring data needs to be imported from the instance
SAP_HolidayCalendar	Use the SAP calendar from an SAP instance to schedule jobs; the calendaring data needs to be imported from the instance
System_Easter	Calculates Easter or Easter-based holidays
System_FiscalPeriods	Calculates open days based on periods in a fiscal year.

You can also define your own period functions, however, since all time windows using a period function share the same instance of a period function, parameter values can be cached for performance reasons. You must be very careful and check your parameters.

Note

You need the scripting module, which requires the *Module.Scripting* license key, to be able to create period functions.

Actions

Period functions support the following actions:

Action	Description
Export	Export the period function into a CAR file
Edit	Edit the period function
Edit Security	Edit the security of the period function
Delete	Delete the period function
Duplicate	Make a copy of the period function to create a similar one
Expand All	Expand all period functions in the current filter
New	Create a new period function
Filter > New Filter	Create a new period function filter
Filter > Edit Filter	Edit current period function filter
Filter > Delete	Delete current period function filter

Easter Period Function

The easter period function accepts one *dayNumber* parameter, which is the offset to easter Sunday.

No parameter or `daynumber=0` means easter Sunday.

SAP Calendars

AP calendars are imported and converted to period functions to be used in time windows. Period functions of SAP calendars take arguments, as the factory calendar, for example, can include several different calendars, which all must have a unique calendar id.

SAP Calendar period functions accept the following arguments:

- calendar=<SAP Instance> <Calendar ID>

Parameter	Description
<SAP Instance>	The SAP Instance Name
<Calendar ID>	The Calendar Id, usually two characters, can be found in the output file of the *SAP_ImportCalendar* job.

You should only specify instances for which you have imported the calendar with the *SAP_ImportCalendar* job definition and the specific calendar ID must exist in that SAP instance. Use the *Opens at* tab of the time window edit screen to make sure you are using the correct values.

Fiscal Period Function

The fiscal period function is used to calculate open days based on the fiscal year, both US and ISO-8601 fiscal years are supported. Fiscal years divide quarters in 1 five- and 2 four-week periods.

US style fiscal years start on January 1st. ISO-8601 fiscal years always start on Monday. The 4-5-4 fiscal period starts on the 2nd workday to the 6th workday of each period.

Parameter	Description	Values	Default
week{1|2|3}	Specifies the number of weeks for the first, second and third period of the quarter	4 (period is 4 weeks long) 5 (period is 5 weeks long)	week1=4 week2=4 week3=5
weekday{From|To}	Specifies the day of the week.	1 (Sunday), 2 (Monday), 3 (Tuesday), 4 (Wednesday), 5 (Thursday), 6 (Friday), 7 (Saturday), 8 (Workday), 9 (Weekends), 10 (Workweek), 0 (Anyday)	weekdayFrom=8 (Workday) weekdayTo=1 (Sunday)
weekOrDay{From|To}	Specifies the number of the weekday in the period; if the number is larger than the number of days in the period, it specifies the last weekday.	integers >=1 (1, 2, 3 ...)	weekOrDayFrom=1 (first day of the period) weekOrDayTo=1 (first day of the period)
isISOPeriod	Specifies the start of the year.	true (Monday) false (January 1st)	false (Year starts on Monday).

Note

The period function parameters are case-sensitive.

Finding Period Functions

You can search for period functions using filters and the *Search Period Functions* box on the Period Functions tab. This box is known as the *intelliSearch* box and located under your username on the top right-hand side of the user interface. Filters allow you to specify a list of objects with static criteria. IntelliSearch allows you to specify complex queries in a simple way using prefixes. Prefixes are used to specify which property you are searching in and have short and long syntaxes. For example, if you want to display all period functions with the term *import* in the comment, you would use the search criteria as follows:

```
c:import
```

You can search more than one property, as follows:

```
c:import n:Bi
```

Note

No spaces should be entered before or after the colon (*:*).

See the Advanced Object Search for more information.

The following table illustrates the available prefixes for period functions:

Prefixes	**Description**
n, name	searches the name property
c, com, comment	searches the comment property
d, desc, description	searches the description property
a, app, application	searches the application property

Deleting Period Functions

You can only delete period functions when no other objects relate to them. For example, if there are time windows that use the period function, the period function cannot be deleted until all time windows that use it have been edited to not use it anymore. You can see all objects that relate to the period function in *Related Objects* in the lower detail pane and on the show page.

The table in related objects contains three columns:

- Type - the type of object with a link to it
- Related Object - the name of the object with a link to it
- Used As - objects can sometimes be used in different roles

Security

Privilege	**Description**
PeriodFunction.Create	Create period functions
PeriodFunction.Delete	Delete period functions
PeriodFunction.Edit	Edit period functions
PeriodFunction.View	Access period functions

You can grant privileges on two levels, *Access* and *Admin;* a privilege granted on *Admin* level allows the grantee to grant the privilege to other users. These privileges can be granted *system-wide*, per *partition* or *isolation group*.

If you have the security module, which requires the *Module.Security* license key, you have an additional *Security* tab on the period function. It allows you to specify which users can access, edit, and delete the period function.

- Granting and Revoking System Privileges
- Granting or Revoking Object Privileges

Example

In the following examples, tables illustrate the fields of the time window element that need to be filled, all other fields are left with their default values.

SAP Calendar Period Function Parameter

In this example, you want to have the time open on the fourth workday of the SAP Factory Calendar of every month. You need 3 time windows in this case. One that is always closed, one that uses the SAP Factory Calendar and one for the day shift.

Note

We need the *TW_AlwaysClosed* and *TW_GermanFactoryCalendar* time windows below because we want to force SAP CPS to use the SAP Calendar for determining and counting workdays, this way it takes public holidays into account.

You need to create a time window that is always closed. Only the following fields need to be specified:

Field	**Value**
Name	TW_AlwaysClosed
Is Calendar	true (checked)

You also need a time window to use the SAP Factory Calendar:

Field	**Value**
Name	TW_GermanFactoryCalendar
Is Calendar	true (checked)

The German factory calendar of the `PR1` instance is used in a time window element as follows:

Field	**Value**
Period Function	SAP_FactoryCalendar
with parameters	`calendar=PR1 FR`

The time window with the day shift is the one you will use in SAP CPS. In this case, it should be open on the first workday of the month and shift 3 days forward when the open day falls on a day where *TW_AlwaysClosed* is closed.

Tab	**Field**	**Value**
Time Window	Name	SAP_4th_Workday
Time Window	Is Calendar	true (checked)

Time Window	If an open day is closed during	*TW_AlwaysClosed*
Time Window	the shift this day	*Forward*
Time Window	to the	3
Time Window	open day in	*TW_GermanFactoryCalendar*
Elements	and from the	1st workday of the month

Easter Period Function

The following time window element defines Good Friday (Friday preceding easter).

Field	**Value**
Period Function	System_Easter
with parameters	`daynumber=-2`

Fiscal Period Function

Field	**Value**
Period Function	System_FiscalPeriods
with parameters	`week1=4 week2=5 week3=4 weekOrDayFrom=2 weekdayFrom=8 weekOrDayTo=6 weekdayTo=8 isISOPeriod=true`

Time Zones

Use

The time zones feature of SAP CPS makes it easy to manage scheduler environments that span multiple time zones. For example, assume there is an office in London. High-load jobs need to be scheduled to run on the server in New York. These jobs run in the night in New York so that the extra server load does not affect the people in the New York office. The job can be submitted to start at 01:00 (for example 1 AM) in the New York time zone and the time offset of London to New York does not need to be calculated. Daylight saving times can be hard to remember, especially when you have to manage multiple time zones on different parts of the globe. SAP CPS now displays Daylight saving time alongside the offset to help you understand the current offset.

The time zone of a job is the time zone specified in the submit wizard. When the job was submitted from RedwoodScript, without specific time zone, the central server time zone is used for the job time zone.

Note

The SAP CPS central server uses the time zone information from the JVM. This data needs to be kept up-to-date for all time zones you use. This is especially true for Daylight Saving Times, which have been changed in various countries in the last years.

Olson Time Zones

SAP CPS uses the Olson time zone format. All time and date calculations are done in the context of a time zone. In most cases this time zone is either simply the session time zone or an explicitly provided time zone.

Arthur Olson started the "America/New York" style time zone definitions that were first put into BSD 4.3. Most if not all Unices now support this style. The official term for them is the "tz" database [http://www.twinsun.com/-tz/tz-link.htm]. The time zones are publicly stored at ftp://elsie.nci.nih.gov/pub and regularly updated. The Olson time zone definitions are a standard reference on what times were historically for the last hundred years.

Time zones are part of the *Environment* section in the navigation bar. The time zones available are loaded directly from the Java Time Zone database when SAP CPS is installed.

Daylight Saving Time (DST)

In the developed world, DST has been introduced to attempt to save energy. For computer systems this means an hour of the day occurs twice or is skipped. SAP CPS has no issues with time switches. If systems in your data-center run SAP CPS jobs and need to be shut-down over the switch period (usually SAP systems), make sure you hold all the affected queues before shutting down the SAP systems, as you do not want new jobs to start. Please wait until all jobs have reached a final state before shutting SAP systems down. Avoid releasing all queues simultaneously, or the SAP systems might get overloaded as soon as they become available. You can also use time windows to achieve the above without human intervention at the time of the date-change.

Actions

Time zones support the following actions:

Action	Description
Edit	Edit the time zone
Delete	Delete the time zone
Expand All	Expand all time zones in the current filter
Filter > New Filter	Create a new time zone filter
Filter > Edit Filter	Edit current time zone filter
Filter > Delete	Delete current time zone filter

It is only possible to edit the description of a time zone (which is used in preference to the name for display purposes). It is not possible to edit any of the other attributes.

Finding Time Zones

You can search for time zones using filters and the *Search Time Zones* box on the Time Zones tab. This box is known as the *intelliSearch* box and located under your username on the top right-hand side of the user interface. Filters allow you to specify a list of objects with static criteria. IntelliSearch allows you to specify complex queries in a simple way using prefixes. Prefixes are used to specify which property you are searching in and have short as well as long syntaxes. For example, if you want to display all time zones with the term *import* in the comment, you would use the search criteria as follows:

```
c:import
```

You can search more than one property, as follows:

```
c:import n:Bi
```

Note

No spaces should be entered before or after the colon (*:*).

See the Advanced Object Search for more information.

The following table illustrates the available prefixes for time zones:

Prefixes	**Description**
n, name	searches the name property
c, com, comment	searches the comment property
d, desc, description	searches the description property

Procedure

1. Choose *Environment* > *Time Zones* and choose a time zone.
2. From the context-menu, choose *Edit*.
3. Enter a description and comment.
4. Choose *Save and Close* to save the time zone.

Time Zone

- *Name* - The name of the time zone.
- *Description* - The description of the time zone.
- *Comment* - A comment about the time zone.

Example

1. Choose *Environment* > *Time Zones* and select the *Europe/Berlin* time zone.
2. From the context-menu, choose *Edit*.
3. Enter *Europe/Walldorf* as description.
4. Choose *Save and Close* to save the time zone.

Importing Objects from SAP Systems

Use

You can import various objects from SAP Systems for use in SAP CPS, imported objects will be used in auto-complete fields, for example job parameters.

This avoids typos in the values. Some objects, like calendars and batch events, can be used in SAP CPS for scheduling. You can create as well as delete or edit imported variants in SAP CPS.

The following objects can be imported from SAP Systems:

Object	**Job Definition**	**Use**	**Topic**
ABAP programs	SAP_ImportAbapPrograms	Used in job parameters as well as for adding, removing and editing parameter values of variants	Importing ABAP Programs
ABAP program parameter definition	SAP_ImportAbapParameterDefinition	Imports the parameter definition of an ABAP program	Importing ABAP Program Parameter Definitions
Archive objects	SAP_ImportArchiveObjects	Used in job parameters	Importing Archive Objects

Batch events	SAP_ImportBatchEvents	Used in job parameters	Importing Batch Events
Calendars	SAP_ImportCalendars	Used for scheduling	Importing Calendars
CCMS jobs	SAP_ImportCcmsJobs	Used for re-scheduling already completed jobs	Importing CCMS Jobs
CCMS monitors	SAP_ImportCcmsMonitors	Monitor multiple SAP Systems from one central location by importing their monitors	Integrating CCMS Monitors
Communication channels	SAP_PI_ImportChannels	Used is job parameters.	Importing Communication Channels
External commands	SAP_ImportExternalCommands	Used in job parameters.	Importing External Commands
Info Packages	SAP_BW_ImportInfoPackages	Used in job parameters.	
Info Package Groups	SAP_BW_ImportInfoPackageGroups	Used in job parameters.	
Java Job Definitions	SAP_JXBP_ImportJobDefinitions	Used to import job definitions from the Java Scheduler	Importing Java Job Definitions
Mass Activities	SAP_MA_ImportMassActivitySAP_MA_ImportMassActivities	Imports one or all Mass Activities, respectively	Importing and Running Mass Activities
Output devices	SAP_ImportOutputDevices	Used in job parameters	Importing Output Devices
Process Chain	SAP_BW_ImportProcessChains	Used in job parameters	Importing and Running Process Chains Restarting Process Chains
Process Chain Definition	SAP_BW_ImportProcessChainDefinition	Used for adding, removing and editing parameter values	see topics above
Recipients	SAP_ImportRecipients	Used in job parameters	Importing Recipients
Variants	SAP_ImportVariants	Used in job parameters as well as for adding, removing and editing parameter values of variants	Importing Variants
Variant Definitions	SAP_ImportVariantDefinitions	Used for adding, removing and editing parameter values of variants	Importing Variant Definitions

See Also

- Default SAP Job Definitions

Importing ABAP Programs

Use

You import ABAP programs to be able to select the correct ABAP when you submit SAP CPS job definitions.

The following job definition allows you to import ABAP programs:

- SAP_ImportAbapPrograms

Parameters

- SAP System - The SAP system you want to import the ABAP's from
- ABAP program - The name or pattern of ABAP program names to import
- Max. count - You can restrict the amount of ABAP programs to import
- Drop old selection? - Allows you to delete previously imported ABAP programs

Procedure

Importing SAP ABAP programs

1. Navigate to *Definitions* > *Job Definitions*.
2. Choose *Submit* from the context menu of the job definition *SAP_ImportAbapPrograms*.
3. Fill the fields and choose *Submit*.
4. Choose *Submit*.

Example

1. Navigate to *Definitions* > *Job Definitions*.
2. Choose *Submit* from the context menu of the job definition *SAP_ImportAbapPrograms*.
3. Select the *SAP System* `PR1`, *ABAP program* `RSUSR*`, leave the other parameters at the default value.
4. Choose *Submit*.

See Also

Handling ABAP programs

Importing ABAP Program Parameter Definitions

Use

You import ABAP program parameter definitions to be able to select the correct ABAP when you submit SAP CPS job definitions.

The following job definition allows you to import ABAP program parameter definitions:

- SAP_ImportAbapProgramParameterDefinition

Parameters

- SAP System - The SAP system you want to import the parameter definitions from
- ABAP program - The name or pattern of ABAP program names of which you want to import parameter definitions
- Overwrite ABAP parameters? - Allows you to overwrite imported ABAP program parameters

Procedure

Importing SAP ABAP program parameter definitions

1. Navigate to *Definitions* > *Job Definitions*.
2. Choose *Submit* from the context menu of the job definition *SAP_ImportAbapProgramParameterDefinition*.
3. Fill the fields and choose *Submit*.
4. Choose *Submit*.

Example

1. Navigate to *Definitions* > *Job Definitions*.
2. Choose *Submit* from the context menu of the job definition *SAP_ImportAbapPrograms*.
3. Select the *SAP System* `PR1`, *ABAP program* `RSUSR*`, leave the other parameter at the default value.
4. Choose *Submit*.

Importing Archive Objects

Use

You import archive objects to be able to select the correct archive object when you submit SAP CPS job definitions.

The following job definition allows you to import archive objects:

- SAP_ImportArchiveObjects

Parameters

- SAP System - The SAP system you want to import the archive objects from
- Drop old selection? - Allows you to delete previously imported archive objects

Prerequisites

- XBP 3.0 must be available in the SAP system

Procedure

Importing archive objects

1. Navigate to *Definitions* > *Job Definitions*.
2. Choose *Submit* from the context menu of the job definition *SAP_ImportArchiveObjects*.
3. Select a target SAP system.
4. Choose *Submit*.

Example

1. Navigate to *Definitions* > *Job Definitions*.
2. Choose *Submit* from the context menu of the job definition *SAP_ImportArchiveObjects*.
3. Select the *SAP System* `PR1`, leave the other parameter at the default value.
4. Choose *Submit*.

Importing Batch Events

Use

You import batch events to be able to select the correct event object when you event history.

The following job definition allows you to import batch events:

- SAP_ImportArchiveObjects

Parameters

- SAP System - The SAP system you want to import the events from
- Drop old selection? - Allows you to delete previously imported events

Prerequisites

- XBP 3.0 must be available in the SAP system

Procedure

Importing SAP ABAP programs

1. Navigate to *Definitions* > *Job Definitions*.
2. Choose *Submit* from the context menu of the job definition *SAP_ImportArchiveObjects*.
3. Select a target SAP system.
4. Choose *Submit*.

Example

1. Navigate to *Definitions* > *Job Definitions*.
2. Choose *Submit* from the context menu of the job definition *SAP_ImportArchiveObjects*.
3. Select the *SAP System* `PR1`, leave the other parameter at the default value.
4. Choose *Submit*.

Importing Calendars

Use

SAP CPS features rich calendaring possibilities; time windows offer great flexibility while remaining easy to define. SAP calendars are imported and converted to period functions to be used in time windows.

The following job definition is used to import SAP Calendars:

- SAP_ImportCalendars

The job definition has the following parameters:

- SAP System - The SAP system to import the calendars from
- Calendar Type - The type of SAP Calendar to import, usually on of *Factory* or *Holiday*.

Procedure

Importing SAP Calendars

1. Navigate to *Definitions* > *Job Definitions*.
2. Choose *Submit* from the context menu of the job definition *SAP_ImportCalendars*.
3. Select an *SAP System* and an *SAP Calendar type*, choose *Submit*.
4. Choose *Submit*.

Importing CCMS Jobs

Use

The SAP Computer Center Management System (CCMS) is part of an SAP solution that provides common functionality such as user management, monitoring and background jobs. You have previously submitted background jobs in SAP via transaction **SM36** and want these jobs to be controlled by SAP CPS. You can import CCMS job steps as separate job definitions or just have the job chains use the *SAP_AbapRun* job definition; the latter keeps the number of job definitions in SAP CPS low and allows you have a better overview. The downside is that you cannot *easily* schedule a step on its own without filling in the parameters yourself or submitting the job chain with steps disabled.

When multiple conflicting jobs match your search criteria, the latest job matching the search criteria will be imported.

Note

You may be restricted to connecting to a single SAP client/instance combination from within SAP CPS at any given time; this depends on the value of the *ProcessServerService.SAP.limit* key in your license and your privileges. If this is the case, please make sure you are in the correct isolation group, corresponding to the SAP client/instance combination where you submitted the jobs initially.

Procedure

To import your CCMS jobs from an SAP System:

1. Choose *Definitions* > *Job Definitions* and locate the job definition `SAP_ImportCCMSjobs`.

2. From the context menu, choose *Submit*.

3. Fill in the corresponding information in both tabs for each parameter (for reference see table below).

4. Choose scheduling on the left-hand side and choose the queue belonging to the SAP System.

5. Choose submit.

Name	**Description**	**Default Expression**
SAP_SYSTEMS	SAP System	In
CLIENT	SAP Client	In Out
JOBNAME	SAP Job Name	In
JOBCOUNT	SAP Job Count	In
SAP_USER_NAME	SAP User Name	In
CLIENT_SPECIFIC	Generate Job Definitions for This Client Only?	In
JOBDEF_NAME	Name of the Generated Job Definition	In
OVERWRITE	Overwrite Existing Job Definitions?	In
OVERWRITE_LANGUAGE	Force Step Language	In
OVERWRITE_USER	Force Step User	In
APPLICATION	Application	In
CREATE_JOBDEFS_FOR_STEPS	Create a Job Definition for Each Step?	In
EXTENDED_PRINT_PARAMETERS	Generate Extra Print Parameters?	In
ADD_STEP_NUMBERS	Add Step Numbers to SAP Job Names?	In
SHOWLOG	Retrieve Job Log?	In
SHOWSPOOL	Retrieve Spool Lists?	In
DELETE_JOB	Delete Job When Finished?	In

Example

You want to import SAP job number `1112`; it was submitted by `jdoe` and executed the ABAP `RSUSR100` in the SAP System `sapprd01`

1. Choose *Definitions* > *Job Definitions* and locate the job definition *SAP_ImportCCMSjobs*.

2. From the context menu, choose *Submit*.

3. Fill in the information below for the *Parameters* tab.

4. Fill in the information below for the *Generation Settings* tab.

5. Choose *Submit* and navigate to *Monitoring* > *Jobs*.

6. Choose *Refresh* to see the newly submitted job.

Parameters tab

- JOBNAME - *RSUSR100*
- JOBCOUNT - 1112

- SAP_USER_NAME - jdoe
- CLIENT - 100
- CLIENT_SPECIFIC - Y

Generation Settings tab

- JOBDEF_NAME - JDOE_RSUSR100
- OVERWRITE - Y
- SHOWLOG - Y
- SHOWSPOOL - Y
- DELETE_JOB - N

See Also

- Configuring an SAP System for Central Process Scheduling
- Connecting to SAP Systems
- Submitting a Job or Job Chain

Importing Communication Channels via PI/XI

Use

You import communication channels to be able to select the correct communication channel when you submit SAP CPS job definitions.

Note

The SAP PI/XI interface requires a Chargeable license.

The following job definition allows you to import communication channels:

- SAP_PI_ImportChannels

Parameters

- SAP System - The SAP system you want to import the archive objects from
- Channel Name - The name of the channel
- Channel Service - The service of the channel
- Channel Party - The channel party
- Drop old selection? - Allows you to delete previously imported archive objects

Prerequisites

- You switched to external control in the Communication Channel Monitor [http://help.sap.com/saphelp_nw04/helpdata/EN/44/2a1a8620323f0ee10000000a114a6b/content.htm].

- You configured SAP CPS to connect to the PI interface of your SAP instance, this is outlined in the Controlling Communication Channels with SAP CPS via PI/XI

Procedure

Importing archive objects

1. Navigate to *Definitions* > *Job Definitions*.
2. Choose *Submit* from the context menu of the job definition *SAP_PI_ImportChannels*.
3. Select a target SAP system and fill in a *Channel Name*.
4. Choose *Submit*.

Example

1. Navigate to *Definitions* > *Job Definitions*.
2. Choose *Submit* from the context menu of the job definition *SAP_PI_ImportChannels*.
3. Select the *SAP System* `PR1` and fill `MyChannel` into the *Channel Name* field; leave the other parameter at the default value.
4. Choose *Submit*.

See Also

- Controlling Communication Channels with SAP CPS via PI/XI
- SAP Communication Channels

Importing External Commands

Use

You import external commands to be able to select the correct archive object when you submit SAP CPS job definitions.

The following job definition allows you to import external commands:

- SAP_ImportExternalCommands

Parameters

- SAP System - The SAP system you want to import the external commands from
- External command name - The name or a pattern of names of external commands
- Operating system - The operating system the commands should belong to
- Max. count - The maximum amount of commands to import
- Drop old selection? - Allows you to delete previously imported external commands

Procedure

Importing external commands

1. Navigate to *Definitions* > *Job Definitions*.
2. Choose *Submit* from the context menu of the job definition *SAP_ImportExternalCommands*.
3. Select a target SAP system, fill in the other parameters.
4. Choose *Submit*.

Example

1. Navigate to *Definitions* > *Job Definitions*.
2. Choose *Submit* from the context menu of the job definition *SAP_ImportExternalCommands*.
3. Select the *SAP System* `PR1`, fill `*` and `UNIX` into the *External command name* and *Operating system* fields, respectively; leave the other parameters at the default value.
4. Choose *Submit*.

Importing Java Job Definitions

Use

As of SAP NetWeaver 7.10, there is a Java Scheduler that allows you to schedule Message-Driven Beans. SAP CPS allows you to connect to the scheduler, import the job definitions and schedule them from within SAP CPS. The Java Scheduler has basic scheduling capabilities, like a cron on Unix or *Scheduled Tasks* on Microsoft Windows, but does not include features such as time windows or complex submit frames. To be able to run jobs against the Java Scheduler you have to connect SAP CPS to the Java Scheduler.

To import job definitions from the Java Scheduler, run the *SAP_JXBP_ImportJobDefinitions* SAP job definition to import multiple job definitions or *SAP_JXBP_GenerateJobDefinition* to import a single one. The imported job definitions can then be scheduled in SAP CPS like any other job.

Procedure

To import all job definitions

1. Navigate to *Definitions* > *Job Definitions*.
2. Choose *Submit* from the context menu of *SAP_JXBP_ImportJobDefinitions*.
3. Select the JXBP SAP System in the drop-down menu.
4. Fill `*` into the *J2EE job definition* field.
5. On *Generation Settings* tab select an application for the job definitions.
6. Choose *Submit*.

Importing and Running Mass Activities

Use

Mass activities can be started and monitored from within SAP CPS like any other job chain. You can specify parameters to override the current parameter set inside your SAP System. You can copy a parameter record, which is a subset of the parameter set in SAP, and override certain parameters. Parameter sets cannot be imported into SAP CPS. Before you run a Mass Activity, a parameter set must be activated for it.

To be able to import Mass Activities, you need to load the ISU transport files, which are available in the *Configuration > Software* section of the user interface.

Note

Scheduling and Monitoring of Mass Activities requires the *ProcessServerService.SAP.MassActivities* license key.

The following default job definitions are available for interacting with Mass Activities:

Job Definition	**Description**
SAP_MA_CopyParameters	Copy mass activity parameters.
SAP_MA_CreateParameters	Create a new mass activity parameter record.
SAP_MA_DeleteDunningProposal	Delete dunning proposal of a new mass activity.
SAP_MA_ImportMassActivities	Import SAP mass activity definitions.
SAP_MA_ImportMassActivity	Import an SAP mass activity definition.
SAP_MA_MassActivityRun	Create a parameter record and start a mass activity run.
SAP_MA_MassActivityTemplateRun	Start a mass activity run using an existing parameter record.
SAP_MA_ResumeMassActivity	Resume processing of an interrupted SAP mass activity run.
SAP_MA_ShowMassActivity	Show SAP mass activity definition.
SAP_MA_StopMassActivity	Stop an SAP mass activity run.

Note

Although the `LAUFD` parameter has a date with time zone data type, only the date part of the value is used.

Prerequisites

- Redwood transport files need to be loaded, see the Enhanced SAP Interfaces with Redwood Transports section of the documentation for more information on loading the transport files.

Procedure

Importing a Mass Activity

1. Navigate to *Definitions > Job Definitions*.

2. Choose *Submit* from the context menu of the job definition *SAP_MA_ImportMassActivity*

3. Fill in the fields and choose *Submit*.

Importing All Mass Activities

1. Navigate to *Definitions* > *Job Definitions*.
2. Choose *Submit* from the context menu of the job definition *SAP_MA_ImportMassActivities*.
3. Fill in the fields and choose *Submit*.

Run a Mass Activity

1. Navigate to *Definitions* > *Job Definitions*
2. Choose *Submit* from the context menu of the job definition *SAP_MA_CopyParameters*.
3. Fill in the fields and choose *Submit*.
    a. Specify the parameter set

Copy a Parameter record from a previous run of a mass activity and run another with that parameter record

1. Navigate to *Definitions* > *Job Definitions*.
2. Choose *Submit* from the context menu of the job definition *SAP_MA_CopyParameters*.
3. Fill in the fields, the source information is used to retrieve the parameter record, the target id will be used when submitting *SAP_MA_MassActivityTemplateRun*.
4. Choose *Submit*.
5. Choose *Submit* from the context menu of the job definition *SAP_MA_MassActivityTemplateRun*.
6. Fill in the fields and choose *Submit*.

Create a parameter set

1. Navigate to *Environment* > *SAP Systems*.
2. Choose *Maintain Objects* from the context menu of the SAP System.
3. Expand *Mass Activities* and choose *New Parameter Set* from the context menu of a Mass Activity.
4. Choose *New Table* or *New Field* and select a value.
5. Repeat previous task for all choose *Save & Close*.

Importing Output Devices

Use

You import output devices to be able to select the correct output device when you submit SAP CPS job definitions.

The following job definition allows you to import output devices:

- SAP_ImportOutputDevices

Parameters

- SAP System - The SAP system you want to import the output devices from

- The output device to import
- The long name of output device to import(XBP 3.0 only)
- Change time stamp (XBP 3.0 only)
- Drop old selection? - Allows you to delete previously imported output devices

Prerequisites

- XBP 2.0 or higher must be available in the SAP system, XBP 3.0 is required if you want to use the *Change time stamp* or *The long name of output device to import* parameters

Procedure

Importing output devices

1. Navigate to *Definitions* > *Job Definitions*.
2. Choose *Submit* from the context menu of the job definition *SAP_ImportOutputDevices*.
3. Select a target SAP system.
4. Choose *Submit*.

Example

1. Navigate to *Definitions* > *Job Definitions*.
2. Choose *Submit* from the context menu of the job definition *SAP_ImportOutputDevices*.
3. Select the *SAP System* `PR1`, leave the other parameter at the default value.
4. Choose *Submit*.

Importing and Running Process Chains

Use

A process chain defines a sequence of processes that are scheduled to wait in the background for an event. These processes can trigger a separate event that can, in turn, start other processes. When you monitor a process chain in SAP CPS, the process chain is displayed like a job chain.

When a process of a process chain reaches the status *Error*, you have the possibility to start just the erroneous process, its parent or the parents' parent, see the Restarting Process Chains section for more information.

Monitoring Process Chains

Note

Monitoring Process Chains requires the *ProcessServerService.SAP.XBPVariant* license key.

You can configure SAP CPS to monitor process chains started from within SAP, they are displayed like job chains and allow you to restart them if you meet the prerequisites for restarting process chains.

See Monitoring Process Chains for more information.

Restarting Process Chains

Note

Restarting Process Chains requires the *ProcessServerService.SAP.ProcessChainRestart* license key.

The following job definitions allows you to start, represent and restart process chains in your BI or BW system.

Job Definition	**Use**
SAP_BW_ImportProcessChains	Used to import process chains
SAP_BW_Process	Represents a process chain
SAP_BW_ProcessChainRun	Used to start a process chain
SAP_CcmsJob	Represents a process chain process

See Restarting Process Chains for more information.

Starting Process Chains with Direct Scheduling in SAP

When you want to schedule process chains in SAP CPS which have *Direct Scheduling in SAP* enabled, make sure the option *Immediate start* is selected. If the *Date/Time* is selected and if the time is in the past, the start of the chain will fail. Note that it will also fail if you try to start it from SAPGui. You need to change the process chains to use either the *Start Immediate* option or change them to *Start Using Meta Chain or API*.

Prerequisites

- An RFC connection has been established between the Central Scheduler and a BI or BW system, as outlined in Connecting to SAP Systems.

Procedure

To import the process chains from an SAP instance

1. Navigate to *Definitions* > *Job Definitions*
2. Choose *Submit* from the context menu of job definition *SAP_BW_ImportProcessChains*.
3. Select the SAP System name of the BW system in the *SAP System* parameter, fill in other parameters if you wish to restrict the number of chains that will be imported.
4. Choose *Submit*.

To start a BW process chain

1. Navigate to *Definitions* > *Job Definitions*
2. Choose *Submit* from the context menu of job definition *SAP_BW_ProcessChainRun*.
3. Select the SAP System name of the BW system in the *SAP System* parameter,
4. Select the process chain to start.
5. Choose *Submit* to schedule the job.

Restarting Process Chains

Use

Process chains can be restarted from within SAP CPS at chain-level as well as on process level. This allows you to only restart the processes that require a restart without starting the whole chain again. When a job of a process chain reaches an error state, the parent job of the process chain is put to console and an operator message is generated allowing you to restart the chain at a desired point. You can choose to make a process chain restartable when you submit the process chain. There are three options:

- the entire chain is restartable
- only failed processes are restartable
- the chain is not restartable

This is governed by the `CHAIN_RESTARTABLE` parameter on the *SAP_BW_ProcessChainRun* job definition. Once you have selected a restart option in the operator message, *SAP_BW_RestartProcess* or *SAP_BW_RestartProcessChain* are submitted for you, depending on your reply to the operator message. To restart process chains, you must first import and run a process chain, this is outlined in the Importing and Running Process Chains section.

Note

Restarting process chains requires the *ProcessServerService.SAP.ProcessChainRestart* license key.

Note

Please refer to SAP Notes 1007419 and 1059049 for further information on required Support Packages.

Process chains can also be restarted automatically, without human intervention. Although it is always best to have human intervention when something goes wrong, your process chain might only fail in a predictable manner, for example, when dependent data has not yet been processed. If this is the case, you can have one or more processes of your process chain restart a number of times after a certain delay, waiting for the data to be processed, for example. Each process can have different settings, delays and maximal restart counts.

Prerequisites

- SAP System with all required Support Packages installed
- Process chains have already been imported into SAP CPS
- The SAP transport files must be loaded, see the Enhanced SAP Interfaces with Redwood Transports section for more information on loading the transport files.

Procedure

Restarting a process chain from the console

1. Expand the top-most parent job of the process chain, which has the status console.
2. Locate the operator message *Restart process chain?* and choose reply from the context menu.
3. Choose a level at which you want to restart the process chain and choose *Save & Close*.

Automatically restarting a process chain

1. Navigate to *Environment* > *SAP Systems*.
2. Choose *Maintain Objects* from the context menu of your SAP System.
3. Expand *Process Chains* and choose the process you would like to restart automatically.
4. Fill in *Max Number Of Restarts* as well as *Restart Delay*, change the *Restart Delay Units* if appropriate.
5. Choose *Save & Close*.

Result

In the first procedure, a copy of the process chain is created in the job monitor, only the selected processes have been restarted; you can locate them under the failed processes.

In the second procedure, the process(es) you configured to restart automatically will be the next time you submit a process chain against that SAP System.

Importing Recipients

Use

You import recipients to be able to select the correct recipient when you submit SAP CPS job definitions.

The following job definition allows you to import recipients:

- SAP_ImportRecipients

Parameters

- SAP System - The SAP system you want to import the recipients from
- SAP Client - The SAP client to use when retrieving the list of recipients
- SAP User Name - The SAP username to use when retrieving the list of recipients
- Distribution list - The SAP client to use when retrieving the list of recipients
- Drop old selection? - Allows you to delete previously imported recipients

Prerequisites

- XBP 3.0 must be available in the SAP system

Procedure

Importing recipients

1. Navigate to *Definitions* > *Job Definitions*.
2. Choose *Submit* from the context menu of the job definition *SAP_ImportRecipients*.
3. Select a target SAP system.
4. Choose *Save & Close*.

Example

1. Navigate to *Definitions* > *Job Definitions*.
2. Choose *Submit* from the context menu of the job definition *SAP_ImportRecipients*.
3. Select the *SAP System* `PR1`, leave the other parameter at the default value.
4. Choose *Save & Close*.

Importing Variants

Use

You import variants to be able to select the correct variant when you submit SAP CPS job definitions.

The following job definition allows you to import variants:

- SAP_ImportVariants

Parameters

- SAP System - The SAP system you want to import the variants from
- SAP Client - The SAP client to use to retrieve the variant
- ABAP Program - The ABAP program the variant belongs to
- ABAP Variant - The variant name or pattern
- Max. Count - the maximum amount of variants to retrieve
- Import Variant Definition? - Allows you to import the variant definition as well, this can also be done separately later
- Drop old selection? - Allows you to delete previously imported variants

Prerequisites

- The ABAP program the variant belongs must have already been imported.

Procedure

Importing variants

1. Navigate to *Definitions* > *Job Definitions*.
2. Choose *Submit* from the context menu of the job definition *SAP_ImportVariants*.
3. Select a target SAP system, an *SAP Client* and select an ABAP program in the *ABAP Program* field.
4. Choose *Submit*.

Example

1. Navigate to *Definitions* > *Job Definitions*.

2. Choose *Submit* from the context menu of the job definition *SAP_ImportVariants*.

3. Select the *SAP System* `PR1` and *ABAP Program* `RSUSR007`, leave the other parameters at the default value.

4. Choose *Submit*.

See Also

- Importing ABAP Programs
- Importing Variant Definitions

Creating and Modifying Variants

Use

Variants are sets of parameter values used for quickly setting ABAP program parameters. You can create and modify variants from within SAP CPS in the SAP System *Maintain Objects* editor screen.

Creating Variants

First you import the ABAP program, then you can create variants. You create variants on the ABAP program and set a number of parameters. You should import at least one variant and its variant definition, so that the available parameters are displayed in drop-down menus. This is done with the *SAP_ImportVariants* job definition, note that it will by default import the variant definition as well. Variant definitions are the set of parameters and their respective values as defined in the variant.

Editing Variants

To be able to edit variants that were created in *SAPGui*, you will have to import the variants and variant definitions first. When you use *SAP_ImportVariants* to import variants, the job will import the variant and its variant definition by default. Variant definitions are the set of parameters and their respective values as defined in the variant.

Saving a Variant in the SAP System for use with SAPGui

You can use the *SAP_StoreVariant* job definition to store variants in you SAP system, see Handling Variants for more information.

Syntax of Select Options

The following operators are available in the Ui, the mathematical syntax can only be used in RedwoodScript and is displayed here for clarity:

Syntax	**Meaning**	**Mathematical Syntax**
NOT	Not	!
EQ	Equals	=
BT	Between	-
NE	Not Equals	!=
NE	Not Equals	<>
GE	Greater than or equals	>=
GT	Greater than	>
LE	Less than or equals	<=

LT	Less than	<

Prerequisites

- You have imported the ABAP program, a variant and its variant definition, as outlined in Importing ABAP Programs and Importing Variants.

Procedure

Create a variant

1. Navigate to *Environment* > *SAP Systems*.
2. Choose *Maintain Objects* from the context menu of the SAP System that represents the SAP Instance where the ABAP program is used.
3. Expand *ABAP Programs* and choose *New Variant* from the context menu of the ABAP program.
4. Fill in a name and optional description, choose *Save*.
5. For each simple parameter/value combination, choose *New Parameter*.
   a. Select the name of the parameter from the drop-down, enter an optional description and value.
6. For each select option, choose *New Select Option*
   a. Select the name of the parameter from the drop-down, enter an optional description.
   b. On the *Values* tab, select to include or exclude the rows in the *Sign* field, select and operator in the *Option* field, and enter a lower and higher value (depending on the operator in *Option*, only a lower value is required).
7. Choose *Save & Close*.

Modify a variant

1. Navigate to *Environment* > *SAP Systems*.
2. Choose *Maintain Objects* from the context menu of the SAP System that represents the SAP Instance where the ABAP program is used.
3. Expand *ABAP Programs*, the ABAP program, client and variant.
4. Select the parameter you want to change.
5. For each simple parameter/value combination"
   a. Select the name of the parameter and enter a value into the *Value* field.
6. For each select option:
   a. Select the name of the parameter.
   b. On the *Values* tab, select to include or exclude the rows in the *Sign* field, select and operator in the *Option* field, and enter a lower and higher value (depending on the operator in *Option*, only a lower value is required).
7. Choose *Save & Close*.

See Also

- Handling Variants
- Creating and Modifying Variants
- Maintaining Variants Manually
- Maintaining Variants Automatically
- Maintaining Variants Automatically (Advanced)

Maintaining Variants Manually

Use

Once you have imported ABAP programs from an SAP instance, you can edit their variants, add, remove parameters and change parameter values inside SAP CPS.

This can be done per SAP System/ABAP/Client/variant combination. If for each ABAP program you import at least one variant as well as its definition, available columns will be presented in a drop-down menu.

The following job definitions allow you to interact with variants:

- SAP_ImportVariants - import variants of already imported ABAP programs
- SAP_ImportVariantDefinitions - import parameter values and select options of variants
- SAP_CopyVariant - copy a variant into another SAP System (you do not have to import the variant)
- SAP_StoreVariant - create/update/replace a variant in an SAP System
- SAP_DeleteVariant - delete a variant from an SAP System

Note

You can only import variants of ABAP programs that have already been imported.

Note

XBP 3.0 requires a logical system for the client, see Setting Up Logical Systems [http://help.sap.com/saphelp_nw70ehp1/helpdata/en/b4/b0b13bb3acef3ce10000000a11402f/frameset.htm] for more information on creating logical systems for clients.

See Creating and Modifying Variants for step-by-step instructions on how to modify variants using the SAP CPS user interface.

Prerequisites

- An RFC connection has been established between the Central Scheduler and a BI or BW system, as outlined in Connecting to SAP Systems.

Additionally, the SAP System must have one of the following:

- XBP 3.0 support

- The Redwood transport files must have been installed

Procedure

Change the value of a variant definition

1. Navigate to *Environment* > *SAP Systems*.
2. Choose *Maintain Objects* from the context menu of the SAP System.
3. Choose an ABAP/client/variant/parameter/select option combination.
4. Change the value of the selected parameter.
5. Optionally, remove a parameter by choosing *Delete* from the context menu of a variant parameter.
6. Choose *Save and Close*.
7. Submit job definition *SAP_StoreVariant* with the parameter values below.

Parameter	**Description**
SAP_SYSTEMS	The SAP System
CLIENT	The client of the SAP System
ABAP_PROGRAM_NAME	The variants ABAP program
ABAP_VARIANT_NAME	Variant
WRITE_MODE	The write mode allows you to specify how the job will handle duplicate entries. The following reactions are available when you encounter a duplicate: Error - the job gets the status error, nothing is changed in the SAP System Skip - the variant is skipped Replace - the variant is overwritten with the variant definition from SAP CPS Change - only different parameters are changed, deleted variant parameters will not be removed (recommended)

Result

The variant in the SAP System has been replaced with the variant that was edited in SAP CPS.

See Also

- Importing Variants
- Importing Variant Definitions
- Creating and Modifying Variants
- Maintaining Variants Automatically
- Maintaining Variants Automatically (Advanced)

Maintaining Variants Automatically

Use

You can use the job definition *SAP_StoreVariant* to change one or more values of a variant directly, without the need to manually edit the variant. You duplicate this job definition and edit the parameter set, the parameter names must match and the values you set must be legal values for the variant. This is especially useful when you want to have dynamic variants, for instance, a parameter that would contain the current date. With SAP CPS, this is easily implemented, as you could schedule a job daily which changes the parameter value of the variant and fills it with the current date. You can also create a job chain, which calls the job definition and sets the values of the variant with parameter mappings. See the Job Chain Job Parameter Mappings section for more information on parameter mappings in job chains.

Note

XBP 3.0 requires a logical system for the client, see Setting Up Logical Systems [http://help.sap.com/-saphelp_nw70ehp1/helpdata/en/b4/b0b13bb3acef3ce10000000a11402f/frameset.htm] for more information on creating logical systems for clients.

Syntax

Variant job parameters should be string parameters, with `PAR_` or `SEL_` prefixes, for parameter and select options, respectively. Their names should contain the name of the parameter/select option in SAP. Select options can optionally have a `_<n>` suffix, when you want to specify more than one. When you want to specify a range, you should enclose it in *square brackets* `[]`, as in `[a-m,!d, !f, !g]`.

The following operators are available for select options, the mathematical syntax can only be used in RedwoodScript and is displayed here for clarity:

Syntax	**Meaning**	**Mathematical Syntax**
NOT	Not	!
EQ	Equals	=
BT	Between	-
NE	Not Equals	!=
NE	Not Equals	<>
GE	Greater than or equals	>=
GT	Greater than	>
LE	Less than or equals	<=
LT	Less than	<

Note

When you specify date-ranges, make sure you put spaces between the dates and the hyphen (-), the range `21.02.2010 - 31.12.2010` is valid and this one is **not valid**:`21.02.2010-31.12.2010`.

Tip

If you want to specify two or more exact values, you can create a parameter for each select option.

For example, you have a job definition with the following parameters:

- `PAR_NOEX` - *Without Extract*, set to `X`
- `SEL_FLDATE` - *Flight Date*, set to `BT` and `21.02.2010 - 31.12.2010`

Another example of a job definition with parameters:

- `PAR_NOEX` - *Without Extract*, set to `X`
- `SEL_CAR` - *Airline*, set to `BT` and `[AA-ZZ, !BC, !AF]`

Alternate example of job definition with parameters:

- `PAR_NOEX` - *Without Extract*, set to `X`
- `SEL_FLDATE_01` - *Flight Date*, set to `GE` and `01.02.2010`
- `SEL_FLDATE_02` - *Flight Date*, set to `LE` and `18.02.2010`

Note the use of two `FLDATE` select options with two exact values.

Prerequisites

- An RFC connection has been established between the Central Scheduler and a BI or BW system, as outlined in Connecting to SAP Systems.

Additionally, the SAP System must have one of the following:

- XBP 3.0 support
- The Redwood transport files must have been installed

Procedure

Duplicate and modify the *SAP_StoreVariant* job definition

1. Navigate to *Definitions* > *Job Definitions*
2. Choose *Duplicate* from the context menu of *SAP_StoreVariant*
3. Choose *Refresh* from the context menu and choose *Edit* from the context menu of *SAP_StoreVariant_1* If there are more than one SAP_StoreVariant_#, edit the one with the highest number.
4. Change the name of the job definition to reflect its purpose.
5. Give *ABAP_PROGRAM_NAME* and *ABAP_VARIANT_NAME* default values.
6. Add additional parameters for the variant parameters that you wish to change. Remember that you can use Redwood Expression Language to specify expressions.
7. Choose *Save & Close*.

Result

You have create a custom job definition that will change a variant on demand, remember that you can change the parameters at submit time again, if necessary.

See Also

- Importing Variants
- Importing Variant Definitions
- Maintaining Variants Manually

- Maintaining Variants Automatically (Advanced)

Maintaining Variants Automatically (Advanced)

Use

You can create a RedwoodScript job definition to change one or more values of a variant directly, without the need to manually edit the variant. In the script you modify the values for the variants with legal values and then submit job definition *SAP_StoreVariant* to save the variant in the SAP System. This is especially useful when you want to have dynamic variants, for instance, a parameter that would contain the current date. With SAP CPS, this is easily implemented, as you could schedule a job daily which changes the parameter value of the variant and fills it with the current date.

Note that you can also use temporary variants which do not require you to run *SAP_StoreVariant* and do not change the variant in the SAP system. The temporary variant is simply deleted after job completion.

Note

XBP 3.0 requires a logical system for the client, see Setting Up Logical Systems [http://help.sap.com/-saphelp_nw70ehp1/helpdata/en/b4/b0b13bb3acef3ce10000000a11402f/frameset.htm] for more information on creating logical systems for clients.

Syntax of Select Options

The following operators are available:

Syntax	Meaning	Mathematical Syntax
NOT	Not	!
EQ	Equals	=
BT	Between	-
NE	Not Equals	!=
NE	Not Equals	<>
GE	Greater than or equals	>=
GT	Greater than	>
LE	Less than or equals	<=
LT	Less than	<

Note

In RedwoodScript, both the `NOT` as well as the `!` syntax can be used.

Prerequisites

- An RFC connection has been established between the Central Scheduler and a BI or BW system, as outlined in Connecting to SAP Systems.

Additionally, the SAP System must have one of the following:

- XBP 3.0 support
- The Redwood transport files must have been installed

Example

The below example shows how to change the *FLDATE* select option of the *KL* variant of ABAP program *BALVEX01* in SAP System *D00*.

Note

The below example does not include the submit of job definition *SAP_StoreVariant*, which can be done in RedwoodScript or in a following step of the job chain.

```
{
// get the SAP System D00
SAPSystem sapSystem = jcsSession.getSAPSystemByName("D00");

// ABAP program BALVEX01
SAPAbapProgram abap = sapSystem.getSAPAbapProgramByName("BALVEX01");

// variant KL in client 200
SAPAbapProgramVariant variant = abap.getSAPAbapVariantByNameClientNumber("KL", "200");

// values of the select option FLDATE
Iterator it = variant.getSAPAbapVariantSeloptByName("FLDATE").getSAPAbapVariantSeloptValues();
if(it.hasNext())
{
  // set the value of the first one to the current date
  String fldate = (String) it.next();
  fldate.setLow(Time.now().toYYYYMMDDString());

  // persist our change
  jcsSession.persist();
}
}
```

See Also

- Importing Variants
- Importing Variant Definitions
- Maintaining Variants Manually
- Maintaining Variants Automatically

Using Temporary Variants

Use

Temporary variants are used to schedule an ABAP program without saving a variant. The variant is created at runtime and deleted once the ABAP program has completed.

There are two ways to use temporary variants:

- You make a copy of one of the *SAP_AbapRun* job definitions and specify the parameters and select options in job parameters on the job definition - these can optionally be modified at submit time.
- You import a variant definition for a given ABAP program, modify it in SAP CPS and submit a job with the ABAP/Variant combination, setting the *Create Temporary Variant?* or `TEMPORARY_VARIANT` parameter to *Y*

Temporary Variants in a Job Definition

By duplicating the job definitions and adding job parameters, you can make use of temporary variants. A temporary variant is created for the job and deleted once the job has finished.

You specify the values and select options by creating job parameters with names matching the technical name of the parameter or select option, with `PAR_` and `SEL_` prefixes, respectively, and setting *Create Temporary Variant?* or `TEMPORARY_VARIANT` to *Y*. You can use the *SAP Object Editor* to inspect previously imported variants for the specific names, see Creating and Modifying Variants in SAP CPS for more information.

Variant Syntax

Variant job parameters should be string parameters, with `PAR_` or `SEL_` prefixes, for parameter and select options, respectively. Their names should contain the name of the parameter/select option in SAP. Select options can optionally have a `_<n>` suffix, when you want to specify more than one. When you want to specify a range, you should enclose it in *square brackets* `[]`, as in `[a-m,!d, !f, !g]`.

Syntax	**Meaning**	**Mathematical Syntax**
NOT	Not	!
EQ	Equals	=
NE	Not Equals	!=
NE	Not Equals	<>
GE	Greater than or equals	>=
GT	Greater than	>
LE	Less than or equals	<=
LT	Less than	<

Example

You have a job definition with the following parameters:

- `PAR_NOEX` - *Without Extract*, set to `X`
- `SEL_FLDATE` - *Flight Date*, set to `BT` and `21.02.2010 - 31.12.2010`

Another example of a job definition with parameters:

- `PAR_NOEX` - *Without Extract*, set to `X`
- `SEL_CAR` - *Airline*, set to `BT` and `[AA-ZZ, !BC, !AF]`

Alternate example of job definition with parameters:

- `PAR_NOEX` - *Without Extract*, set to `X`
- `SEL_FLDATE_01` - *Flight Date*, set to `GE` and `01.02.2010`
- `SEL_FLDATE_02` - *Flight Date*, set to `LE` and `18.02.2010`

Note the use of two `FLDATE` select options with two exact values.

Prerequisites

- An SAP System that features XBP 3.0 or has the latest Redwood transport files installed

Procedure

Use an edited Variant in *SAP_AbapRun* as temporary variant

1. Navigate to *Definitions* > *Job Definitions*
2. Choose *Submit* from the context-menu of the job definition *SAP_AbapRun*.
3. Fill in *SAP System*, *SAP Client*, *SAP Job Name*, *ABAP Program*, *ABAP Variant*, and *Create Temporary Variant?* to *Y*.
4. Choose *Next* two times, set a *Queue* and choose *Submit*.

Note

The edited variant will only be used if *Create Temporary Variant?* is set to *Y*, if you do not set that parameter, the variant definition in the SAP system will be used.

Use job definition parameters to specify parameters and select options

1. Navigate to *Definitions* > *Job Definitions*
2. Duplicate the *SAP_AbapRun* job definition.
3. Choose *Edit* from the context menu of the newly created job definition.
4. On the *Parameters* tab, specify the following:
   a. `PAR_<parameter_name>` for parameters
   b. `SEL_<select_option_name` for select options
5. Choose *Save & Close*.
6. Choose *Submit* from the context menu of the newly edited job definition.
7. Set the *Create Temporary Variant?* to *Y* and fill other required parameters.
8. Choose *Submit*.

See Also

- Handling Variants
- Creating and Modifying Variants
- Handling ABAP Programs
- Maintaining Variants Manually
- Maintaining Variants Automatically
- Maintaining Variants Automatically (Advanced)
- Importing Variants
- Importing Variant Definitions

Redwood Expression Language

Use

Redwood Expression Language (or REL) is a simple expression language that supports basic arithmetic and comparison operators, and some predefined functions. Different scripting contexts have different predefined objects, the functions on this page are available in all contexts. Not all functions are listed here, see REL Functions for a complete list of functions.

You can create your own functions for use in REL and RedwoodScript using a library with REL entry points, see Libraries for more information.

The REL can be used to specify:

- Job definition parameter default values
- Job definition return code mappings
- Job chain step preconditions
- Job chain job preconditions
- Job chain job parameter mappings
- Job chain job scheduling parameters

Expressions in REL are designed to be very similar to expressions in RedwoodScript, and the predefined functions are the same.

Operators

Redwood Expression Language provides support for the basic arithmetic operators:

- + - addition and string concatenation
- - - subtraction
- * - multiplication
- / - division
- % - integer modulus

It also supports comparison:

- === - equal to
- !== - not equal to
- > - greater than
- >= - greater than or equal
- < - less than
- <= - less than or equal

Escape Character

- \ - escape character, and thus must be escaped if the literal is meant to be used. This means that a UNC path would be specified as follows:

  ```
  \\\\server\\share\\folder\\file
  ```

.

Predefined functions

A number of predefined functions are available, these are grouped into functional areas.

Math

- Math.abs(*x*) - take the absolute value of a number. For positive numbers, this is the number, for negative numbers this is the opposite of the number. For example: Math.abs(1) = 1, Math.abs(-10) = 10.
- Math.floor(*x*) - take the floor of a number, this is the largest integer less than or equal to x. For example: floor(2.9) = 2, floor(-2) = -2 and floor(-2.3) = -3
- Math.ceil(*x*) - take the ceiling of a number, this is the smallest integer not less than x. For example: ceiling(2.3) = 3, ceiling(2) = 2 and ceiling(-2.3) = -2.
- Math.round(*x*) - round x to the nearest integer. This uses the 'bankers rounding' algorithm to round numbers .5 exactly, such numbers are rounded to the nearest even number. For example: Math.round(2.2) = 2, Math.round(2.4) = 2, Math.round(3.5) = 4.

Range

- Range.inRange(*candidate*, *range*) - *is* candidate *in the range specified by* range". For example: Range.inRange(2, '1-3') = true, Range.inRange(4, '2-3,6-') is false.
- Range.inSet(*candidate*, *set*) - *is* candidate *in the range specified by* set". For example: Range.inSet(1, ' X X X X ') = true, Range.inSet(6, '_1_3_5_7') = false.

Ranges are made up of components, separated by commas. Each range component can be:

- x - the number *x*
- -x - all numbers from zero to *x* (including *x*).
- x-y - all numbers from *x* to *y* (including both *x* and *y*)
- y- - all numbers from *y* to the maximum value for an integer.

Example ranges:

- *1-5,7-9*
- *-5, 8-*
- *1,2,4,8,16*

Sets are made up of elements, starting at zero. If the element at position x is ' ' (space) or '_' (underscore) then that element is NOT in the set. Otherwise that element is in the set.

Example sets:

- *0 2 4 6* - contains 0, 2, 4, 6
- *X X X X* - contains 0, 2, 4, 6
- *_X_X_X_* - contains 1, 3, 5

Example expression:

- =Range.inRange(1, '0-5') is true
- =Range.inSet(2, '01_345') is false

Time

Function	**Arguments**	**Description**
Time.now()	timeZoneId	The time in the specified time zone; for example (2011/08/23 09:39:33,324 GMT)
Time.now()		The time in the specified time zone; for example (2011/08/23 09:39:33,324 GMT)
Time.isTimeWindowOpen()	DateTimeZone, timeWindowName	Is the specified time window open at the specified time.
Time.isTimeWindowOpenNow()	timeWindowName	Is the specified time window open at the current system time.
Time.expression()	DateTimeZone, expression	Apply expression to DateTimeZone
Time.expressionNow()	expression	Apply expression to Time.now()

Time expressions are a sequence of one or more operations, applied in order to a time (generally now, but may be a specified date).

The following time expressions are available:

- set <specifier> <value> - set <specifier> to <value>
- add <value> <specifier> - add <value> to <specifier> (This may propagate)
- subtract <value> <specifier> - subtract <value> from <specifier> (This may propagate)
- truncate <specifier> - truncate at <specifier> (This will zero <specifier> and everything below it).

The <value> is always a number:

- The days of the week start at 1 for Sunday.
- The days of the year start at 1 for 1 January.

The <specifier> is one of:

- add/subtract: second, minute, hour, day, week, month, year
- truncate: second, minute, hour, day, month
- set: second, minute, hour, day, dow, day_of_year, week, week_of_month, month

The *day* is always the day of the month, and *dow* is the day of the week.

Note

The date you enter is treated in a lenient fashion, that is `32/12/2011` is treated as `01/01/2012`.

Example time expressions:

- add 1 minute
- add 3 seconds
- set hour 1
- truncate day
- subtract 2 days

Example using the Time functions

In these examples, the current time is 2011/08/23 10:50:44,910 GMT according to output of the `=Time.now('GMT')` function.

Example 1

```
=Time.isTimeWindowOpen(Time.now('GMT'), 'System_Week_WorkDays')
```

This expression returns true if at evaluation-time it is a workday in the GMT time zone. An expression like this might be used as a job chain step (or job) precondition to prevent certain steps from running on weekends.

Example 2

```
=Time.expressionNow('add 2 hours add 1 minute add 1 second subtract 2 days')
```

This expression added 2 hours, 1 minute a second and subtracted 2 days from the current time and returned `2011/08/21 13:51:45,910 Europe/Paris`

Example 3

```
=Time.expression(Time.now('GMT'), 'truncate minute')
```

This expression returned: `2011/08/23 10:50:00,000 GMT`

Example 4

```
=Time.expression(Time.now('GMT'), 'set minute 13')
```

This expression returned: `2011/08/23 10:13:44,910 GMT`

Example 4

```
=Time.expression(Time.now('GMT'), 'subtract 1 month')
```

This expression returned: `2011/07/25 10:13:44,910 GMT`

Example using the Time functions as a precondition:

```
=Time.isTimeWindowOpen(Time.now('GMT'), 'System_Week_WorkDays')
```

The function returned *false*, so the job is skipped

Strings

String functions are all applied to a string instance. Strings are treated as arrays of characters, with the first index at 0 and the final index at length-1.

- aString.concat(*s1*, [*s2*[, ...]]) - concatenate *s1*, *s2*, ... to *aString*. This function takes one or more arguments. For example: 'a'.concat('b') = 'ab', 'a'.concat('b', 'c', 'd') = 'abcd'
- aString.charAt(*pos*) - return the character at *pos* in *aString*. The index *pos* is zero based. For example: '012345'.charAt(2) = '2'.
- aString.indexOf(*candidate*[, *startIndex*]) - return the first index of *candidate* in *aString*. If *startIndex* is not specified, the search starts at 0, otherwise it starts at *startIndex*. For example: '012345345'.indexOf('3') = 3
- aString.lastIndexOf(*candidate*[, *startIndex*]) - return the last index of *candidate* in *aString*. If *startIndex* is not specified, the search starts at 0, otherwise it starts at *startIndex*. '012345345'.lastIndexOf('3') = 6
- aString.split(*separator*, *limit*) - slice *aString* using separator, returning at most *limit* entries. For example: '1,2,3,456,7'.split(',', 4) = ['1', '2', '3', '456,7']
- aString.substring(*start*[, *end*]) - return a substring of *aString* starting at *start*. If *end* is not specified the remainder of the string is returned, otherwise it is the last index to return. '01234'.substring(2,4) = '234'
- aString.toLowerCase() - convert the string to lower case. For example: 'AbC'.toLowerCase() = 'abc'
- aString.toUpperCase() - convert the string to upper case. For example: 'aBc'.toUpperCase() = 'ABC'

Note

Like in Java, the + (plus) character can be used for concatenation of strings. For example `'Hello '+'Universe!'`

Casts

Change a data type to another.

- ToBoolean(*x*) - convert *x* to a boolean.
- ToNumber(*x*) - convert *x* to a number.
- ToInteger(*x*) - convert *x* to an integer.
- ToString(*x*) - convert *x* to a string.

Note

Expressions always start with an equals sign (=).

Constraints

The following functions are available for validating parameter values. You specify these functions in the *Simple Constraint Data* field of a parameter; note that the *Simple Constraint Type* must be set to *Expression* for these functions to work.

Constraint.listConstraint(titles, list, bool)

Constraint.pairListConstraint(titles, list, bool)

- titles is a comma separated list of titles.
- list is either a normal list or a pair list
- bool is a boolean expression involving the parameter value

```
=Constraint.listConstraint('Country', 'de,nl,gt', value !== '')
=Constraint.pairListConstraint('Country, Tel. Code', 'de=49,nl=31,gt=502', value !== '')
```

Tables

The following functions are declared within the `Table` object and allow you to interact with tables, such as the *System_Variables* table.

- getColumnString(table, key, column)
- getRow(table, key)
- getColumnNumber(table, key, column)
- formatRow(table, key, rowStart, columnFormat, columnSeparator, rowEnd)
    - *columnFormat* - accepts strings as well as the following substitution parameters to specify the position of the elements:
        - *{0}* - column header
        - *{1}* - value
        - *{2}* - columnSeparator (should always be the first)

The following examples use the *Countries* table

Countries Table:

Key	**Name**	**Abbreviation**	**Translation**	**Conversion_Rate**
31	The Netherlands	NL	Nederland	1.0
33	France	Fr	France	1.0
49	Germany	De	Deutschland	1.0
502	Guatemala	Gt	Guatemala	11.6

Note

The *Key* column is a special built-in column; it is used by the table functions, for example, to look up data in the table and does not need to be specified in the table definition.

getColumnString

```
=Table.getColumnString('Countries', '31', 'Translation')

Returns:
Nederland
```

getColumnNumber

```
=Table.getColumnNumber('Countries', '502', 'Conversion_Rate')

Returns:
11.6
```

getRow

```
=Table.getRow('Countries', '33')
Returns:
FrFranceFrance
```

formatRow

```
=Table.formatRow('Countries', '49', '#', '{2}{0}={1}', '|', '#')

Returns:
#Abbreviation=De|Conversion_Rate=1.0|Name=Germany|Translation=Deutschland#
```

```
=Table.formatRow('Countries', '49', '||', '{2}{1}', '#', '||')

Returns:
||De#1.0#Germany#Deutschland||
```

```
=Table.formatRow('Countries', '49', '', '{2}{1}', '\n', '')

Returns:
De
1.0
Germany
Deutschland
```

The following Redwood Expression Language function allows you to retrieve values out of the built-in *System_Variables* table:

- Variable.getString(key)

It is a shorthand for `Table.getColumnString('System_Variables', key, 'SystemValue')`

Queries

Availability: Requires the Chargeable version

The following functions allow you to query the database, these functions do not return iterators, only the first value of the first column.

- Query.getString(query string, [bind variables])
- Query.getNumber(query string, [bind variables])

```
=Query.getString('select Job.Description from Job where Job.Description = ?', ['GL_REP_EMEA'])
```

```
=Query.getNumber("select Job.JobId from Job where Job.Description = GL_REP_EMEA')
```

The following function returns a job object, it is to be used in a job chain.

Job chain job names take the form Step name, job job number. For example: "Step 2, job 3" or "Data load, job 1".

Query.getRelativeJob(jobname string)

```
Job jcJob = Query.getRelativeJob("Data load, job 2");
```

JobChainParameters

- chainParameters.<parameter_name>
- chainOutParameters.<parameter_name>
- chainJobId
- JobChainParameters.getOutValueNumber(String jobName, String parameterName)
- JobChainParameters.getOutValueString(String jobName, String parameterName)
- JobChainParameters.getOutValueDate(String jobName, String parameterName)

JobChainParameters.getOutValueString('backup, Job 1', 'connectString')

Job chain job names take the form Step name, job job number. For example: 'Step 2, job 3' or 'Data load, job 1'.

Get the value of the Date output parameter bckDate of the job named *Backup, Job 1* in the current job chain, which is, by default, the first job of the step named *Backup*.

Availability: Requires the Chargeable version

```
JobChainParameters.getOutValueDate('Backup, Job 1', 'bckDate')
```

Get the value of the Number output parameter BckSize of the job named *Backup, Job 1* in the current job chain, which is the first job of the step called *Backup*.

Availability: Requires the Chargeable version

```
JobChainParameters.getOutValueNumber('Backup, Job 1', 'bckSize')
```

Get the value of the String output parameter connectString of the job named *Backup, Job 1* in the current job chain, which is the first job of the step called *Backup*.

Availability: Requires the Chargeable version

```
JobChainParameters.getOutValueString('backup, Job 1', 'connectString')
```

See Also

- Scripting contexts
- RedwoodScript

RedwoodScript

Use

RedwoodScript is available with the Scripting module, which requires a specific license; please check the License section for more information. It is a superset of the Redwood Expression Language and can be used in different scripting contexts.

RedwoodScript is a Java-like language based on Java 1.4.2. It allows you to script the repository using an extensive API, which is documented in a separate document available in combination with a Scripting Guide. If you have the Scripting Module but no API documentation and/or no Scripting Guide, please contact Redwood Software.

Note

It is important to follow guidelines and conventions [http://java.sun.com/docs/codeconv/html/-CodeConvTOC.doc.html] when writing code! This makes it easier for you and others to understand your code and will save you a lot of time when you will have to change something, even in your own code, 6 months from now.

When you have the Scripting license, you can use RedwoodScript (expressions) in:

- Actions
  - Job definition on-change
  - Job definition pre-running
  - Job definition post-running
- Triggers
  - Before Definition Change
  - Before Job On Change
  - Before Job On Change Maintenance
  - Before Job Pre Running
  - Before Job Post Running
  - On Job File Content Access
- Job definitions using the RedwoodScript job definition type.
  - RedwoodScript in Jobs
- The Shell found in the user interface under *Scripting*.
  - RedwoodScript in Queues

- RedwoodScript in Process Servers

Most contexts have predefined objects, see the Scripting Contexts section for more information.

The following functions and methods are available in RedwoodScript:

- A subset of Java classes.
- Redwood Expression Language functions.
- Redwood API.

For security reasons, not all Java classes are available to RedwoodScript code. You can import the following classes using `importPackage()`:

- `java.lang.Byte`
- `java.lang.Char`
- `java.lang.Short`
- `java.lang.Integer`
- `java.lang.Long`
- `java.lang.Number`
- `java.lang.Math`
- `java.lang.StrictMath`
- `java.lang.Float`
- `java.lang.Double`
- `java.lang.String`
- `java.lang.Boolean`
- `java.util.*` (but not subpackages of `java.util`)
- `java.math.*` (including subpackages)
- `java.text.*` (including subpackages)

Syntax

RedwoodScript provides three syntax options:

- Short
- Short with imports
- Full

The first two are abbreviated syntax that allow you to dispense with all or most of the boiler plate code that is normally required to write RedwoodScript code that interacts with SAP CPS. The boiler plate includes all the declarations you would usually have to make yourself.

The *Short* mode automatically generates everything, including import statements for classes that you are most likely to use in your task. This mode is activated if the first symbol is `{`.

The *Short with imports* mode allows you to specify the imports. In this case there are no *automatic* imports, and you will have to specify all the imports you need. This mode is activated if the first symbol is `import`.

The *Full* mode lets you write the entire Java class. This mode is activated if the first symbol is `package`. In full mode you will have to explicitly extend the stub that is provided for you. This will be done automatically in all other modes.

Note

RedwoodScript code in the library source field must always be in *Full* mode.

The following illustrates the use of the *short* syntax for a *Hello World!* program:

```
{
  jcsOut.println("Hello, World!");
}
```

The following illustrates the "Hello, World" example when it is fully expanded into RedwoodScript:

```
package com.redwood.scheduler.custom;

import java.math.BigDecimal;
import java.util.Iterator;
import com.redwood.scheduler.api.date.DateTimeZone;
import com.redwood.scheduler.api.model.*;
import com.redwood.scheduler.api.model.enumeration.*;

public class name
extends nameStub
{
  public void execute()
  throws Exception
  {
    jcsOut.println("Hello, World");
  }

   // ... Lots of code ...

}
```

As you can see there is quite a considerable amount of code generated for you. The only difference between the *short* and *short with imports* is that in *short* mode, all the imports are specified for you, and in *short with imports* mode *you* must specify *all* required imports yourself.

Redwood API

In most contexts, frequently used Redwood API classes are imported by default, the imported classes and objects are visible in the Stub code.

You can find information about objects and their types in `apidocs.zip`, which is provided when you have purchased the Chargeable version. If you extract this, and open apidocs/index.html you will be taken to the main index of the API documentation.

The main index consists of two sections:

- Packages - the top table, in the form of JavaDoc.
- The Scheduler API and Language Reference - documentation about context variables and programming using RedwoodScript.

Please look at each of the links in the *Scheduler API and Language Reference* at the bottom of the page carefully, they contain the answers to many *frequently asked questions*.

The *Data model* shows you the available tables in the database for use with RedwoodScript, see the *Querying the database* section below for examples of how to query the database with RedwoodScript.

Object Model

Repository objects have types, for example:

- Job
- JobDefinition
- Queue
- TimeWindow

The type of the object determines the methods you can call.

You call methods on an instance of an object. The available methods are documented in the API documentation.

You can get repository objects from a predefined object called the *jcsSession* which is an instance of *SchedulerSession* in the API documentation:

```
[...]
JobDefinition jobDefinition = jcsSession.getJobDefinitionByName("System_Info");
[...]
```

Once you have an object, you use methods to get information from it, or to manipulate it.

```
[...]
 jcsOut.println(jobDefinition.getName());
[...]
```

Submitting a job

```
{
//Get the job definition
JobDefinition jobdef=jcsSession.getJobDefinitionByName("System_Info");

//Create the Job
Job infoJob=jobdef.prepare();

//Get the Queue
Queue SystemQ=jcsSession.getQueueByName("System");

//Attach queue to job
infoJob.setQueue(SystemQ);

//Print out the jobid of the job
jcsOut.println(infoJob.getJobId());
//Submit the job
jcsSession.persist();
}
```

More documentation is available in the API documentation and on the Internet, RedwoodScript being a Java-like language, there are a lot of Java resources online, remember that you must look at Java 1.4, as 1.5 and 1.6 implement a lot of shorthands which are not available in SAP CPS.

Querying the Database

You cannot query the tables in the database directly, only an upper-layer table-implementation; the available tables in SAP CPS are listed in the API documentation. The Data model and can be found under the Scheduler API and Language Reference section on the index page.

The database can be queried with the following function, which is defined in the SchedulerSession object:

```
executeQuery(query string, [[bind variables])
```

Standard SQL can be used in queries, however, when referring to a column, you need to prepend the table name, which name is the type of the object you want to query for, followed by a period (.), as shown below;

```
jcsSession.executeQuery("select Job.JobId,Job.Description from Job", null, dest);
```

The output can be generated in HTML (for output on screen), CSV (comma separated values, for spreadsheet applications) and XML. For this you need to define a `Reporter` and `ReportDestination;` both require their own classes. The code below illustrates how to create a `Reporter`, a `ReportDestination;` and how to use a query for the report.

```
import com.redwood.scheduler.api.model.report.Reporter;
import com.redwood.scheduler.api.model.report.ReportDestination;
{
  String query = "select Job.JobId,Job.Description from Job where Job.JobId <= 244";
  Reporter reporter = jcsSession.createReporter(jcsOut);
  ReportDestination dest = reporter.getCSVReportDestination();
  jcsSession.executeQuery(query, null, dest);
}
```

The same example as above using bind variables:

```
import com.redwood.scheduler.api.model.report.Reporter;
import com.redwood.scheduler.api.model.report.ReportDestination;
{
  String query = "select Job.JobId,Job.Description from Job where Job.JobId <= ?";
  Reporter reporter = jcsSession.createReporter(jcsOut);
  ReportDestination dest = reporter.getCSVReportDestination();
  jcsSession.executeQuery(query, new Object[] { new Long(244L) }, dest);
}
```

Bind variables can increase the performance and scalability of queries, especially simple queries, like the one above. Some supported databases, like Oracle, parse every query once and skip some queries they have already parsed. So if the Job.JobId above changes frequently, the database will have to parse each and every query for each Job.JobId, if we use a bind variable, this step can be skipped.

The following example is for illustration purposes, when you do not know the job status code and want to use it in a query (note that looking up the code is easier). You have to import the `JobStatus` class, the fully qualified class name can be found in the API documentation:

```
import com.redwood.scheduler.api.model.enumeration.JobStatus;
import com.redwood.scheduler.api.model.report.Reporter;
import com.redwood.scheduler.api.model.report.ReportDestination;
{
  String query = "select Job.JobId,Job.Description from Job where Job.Status = ?";
  Reporter reporter = jcsSession.createReporter(jcsOut);
  ReportDestination dest = reporter.getCSVReportDestination();
  jcsSession.executeQuery(query, new Object[] { String.valueOf(JobStatus.ScheduledCode) }, dest);
}
```

Dates

Dates are lenient, that means that `32/12/2011` is treated as `01/01/2012`.

Example:

November only ever has 30 days. In this first example, the resulting date is: `2011/12/01 12:10:10,010 Europe/Paris` (the time zone in this example is *Europe/Paris*)

```
{
   DateTimeZone dtz = new DateTimeZone(2011,10,31,12,10,10,10);
   jcsOut.println(dtz.toString());
 }
```

Comparing Timestamps

Dates or timestamps in the repository are stored as a combination of an offset since 'the start of time' and a time zone. The start of time (also known as the *epoch*) is the number of milliseconds since midnight, Jan 1st 1970 UTC.

To compare dates in queries, a SQL function named `TODATE` can be used:

- `TODATE(input, timezone)` - Use default format (yyyy-MM-dd HH:mm:ss)
- `TODATE(input, timezone, format)` - Use Java SimpleDateFormat [http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html]

The following code illustrates comparing two dates:

```
import com.redwood.scheduler.api.model.report.Reporter;
import com.redwood.scheduler.api.model.report.ReportDestination;
{
  Reporter reporter = jcsSession.createReporter(jcsOut);
  ReportDestination dest = reporter.getCSVReportDestination();
  jcsSession.executeQuery("select JobId from Job where CreationTime > TODATE('2010-09-01', 'CET', 'yyyy-MM-dd
}
```

Using RedwoodScript in Objects

Object	Topic
Jobs	RedwoodScript in Jobs

Queues	RedwoodScript in Queues
Process Servers	RedwoodScript in Process Servers

Scripting in the Shell

Use

The Shell allows you to access the Repository directly with a scripting language. You can use it to automate repetitive tasks, for example, submit 10 jobs. It can also be used to write or debug your RedwoodScript or Redwood Expression Language code as well.

Note

The shell requires the Scripting module.

By default, the scripting locale is set to *English*, you can change it to *German* using the following code:

```
importPackage(java.util);

jcsSession.setLocale(Locale.GERMAN);
```

Procedure

1. Navigate to *Scripting* > *Shell*.
2. Enter your code in the lower pane.
3. Choose *Submit Command*.
   a. The result of your code will be printed in the upper pane.
4. Close the tab when you are finished.

Example

To submit one job without parameters, you could use the following code:

```
{
 // Get the job definition object
JobDefinition jobDefinition = jcsSession.getJobDefinitionByName("System_Info");

 // Create a job object
Job job = jobDefinition.prepare();

// To submit the job, you need to commit changes to the database
jcsSession.persist();
}
```

To submit 10 jobs without setting any parameters, you could use the following code:

```
{
 // Get the job definition object
JobDefinition jobDefinition = jcsSession.getJobDefinitionByName("System_Info");

for (int i = 0; i < 10; i++)
```

```
 {
 // Create a job object
   jobDefinition.prepare();
 }

 // To submit the jobs, you need to commit changes to the database
jcsSession.persist();
}
```

Triggering Custom Code

Use

Triggers and actions allow you to interact with changes that occur in the system. You can trigger code to run in-between jobs, to prepare the environment, for example, or react on a job change or status with triggers and job definition actions. The main difference between the two is the scope. Actions are limited to the object, and triggers are global and apply to all job definitions (*Before Job* triggers) or all objects (*Before Definition Change* trigger).

When you want to trigger code for jobs, you should decide which solution to implement based on the amount of affected job definitions and the total amount of job definitions that you use regularly. Triggers are a little more expensive in terms of performance, because they fire for every job. Actions require you to specify them on every affected job definition.

1. Using Triggers
2. Using Actions
    a. Customizing Jobs with Actions
3. Executing Your Own Classes

Using Triggers

Use

The following trigger-points are available:

- Before Definition Change - before the definition of an object is changed
- Before Job On Change - when a job changes its status (except for the status change from `Queued` to `lockWait`)
- Before Job Pre Running - before the PreRunningAction code is executed
- Before Job Post Running - before the PostRunningAction code is executed

> **Note**
>
> The *Before Definition Change* trigger-point does not apply to any change to a job, operator message, event or alert. It does not apply to changes in queue status or process server status, for example or to changes made by the system.

You need to understand the concepts of job states to get a clear understanding of when triggers and actions fire, see the Job States section for more information on the subject.

You can have multiple triggers for one trigger point, the sequence in which the triggers are executed inside a trigger-point are defined in the *Execution Order* field.

The code you write must be valid RedwoodScript, however only a subset of Java classes are available for use in triggers. You import your own class from your library for use in a trigger. Redwood recommends to store as much of the code as possible in a library, especially if you plan on reusing it in other triggers or actions.

Triggers have their own log files. These are used only when enabled and accessible via the context menu.

Note

You must never persist a session in your trigger code, that is done for you at a later point.

Trigger Executions

The *Before Job On Change* trigger fires at least twice, once when switching from *New* to *Scheduled* and again from *Scheduled* to *Queued*. Other job-related triggers might fire several times depending on your code. Note that you can create endless loops in triggers, and that these can have severe consequences as many jobs can be affected. Test your triggers well and always start by excluding as many jobs as possible as early as possible (for performance reasons).

When you set a new queue on a job in a *Before Job Pre Running* trigger, the trigger will fire again for the change to the new queue. This can be dangerous, especially if you do not make sure the trigger will not change the queue the second time the code executes for that job as this can end in an endless loop.

Actions

Triggers support the following actions:

Action	Description
Export	Export the trigger into a CAR file
Edit	Edit the trigger
Edit Security	Edit the security of the trigger
Delete	Delete the trigger
Duplicate	Make a copy of the trigger to create a similar one
Expand All	Expand all triggers in the current filter
New	Create a new filter
Filter > New Filter	Create a new trigger filter
Filter > Edit Filter	Edit current trigger filter
Filter > Delete	Delete current trigger filter

Writing Trigger Code

When you write trigger code, you have the choice between creating your own class and writing directly between curly brackets `{}`.

If you choose to surround your method with curly brackets, it will be added to the `execute()` method of the stub class.

Your own class must:

- inherit from the stub

- must implement a `public void execute()`

Finding Triggers

You can search for triggers using filters and the *Search Triggers* box on the Triggers tab. This box is known as the *intelliSearch* box and located under your username on the top right-hand side of the user interface. Filters allow you to specify a list of objects with static criteria. IntelliSearch allows you to specify complex queries in a simple way using prefixes. Prefixes are used to specify which property you are searching in and have short and long syntaxes. For example, if you want to display all triggers with the term *trigger* in the comment, you would use the search criteria as follows:

```
c:trigger
```

You can search more than one property, as follows:

```
c:trigger n:JV
```

Note

No spaces should be entered before or after the colon (*:*).

See the Advanced Object Search for more information.

The following table illustrates the available prefixes for triggers:

Prefixes	Description
n, name	searches the name property
c, com, comment	searches the comment property
d, desc, description	searches the description property
a, app, application	searches the application property

Security

Privilege	Description
Trigger.Create	Create triggers
Trigger.Delete	Delete triggers
Trigger.Edit	Edit triggers
Trigger.View	Access triggers

You can grant privileges on two levels, *Access* and *Admin;* a privilege granted on *Admin* level allows the grantee to grant the privilege to other users. These privileges can be granted *system-wide*, per *partition* or *isolation group*.

If you have the security module, which requires the *Module.Security* license key, you have an additional *Security* tab on the trigger. It allows you to specify which users can access, edit, and delete the trigger.

Contexts

For each trigger point, a set of objects are predefined in the context for you to use. Once you have chosen your trigger-point, you can refresh the *Stub Source* from the context menu to display the list of predefined objects with their classes (so that you can use the API documentation to lookup the available methods). The objects *jcsOut* and *jcsOutLog* write to the same file, the latter uses the Redwood Logging Syntax, which allows you to specify a severity (info, warning, error, fatal, debug) and includes a time-stamp.

The following tables illustrate the predefined objects, their classes as well as a simple example.

Before Definition Change

Object	Class	Example Code
jcsOutLog	com.redwood.scheduler.infrastructure.logging.api.Logger	jcsOutLog.fatal("This is a fatal error message!");
jcsTriggerContext	com.redwood.scheduler.api.scripting.variables.TriggerScriptObject	jcsTriggerContext.getSchedulerEntity();

Before Job On Change

Object	Class	Example Code
jcsOutLog	com.redwood.scheduler.infrastructure.logging.api.Logger	jcsOutLog.warn("This is a warning and gets printed to the stdonchangelog file on the job.");
jcsOnChangeContext	com.redwood.scheduler.api.scripting.variables.PreExecutingActionScriptObject	jcsOnChangeContext.setFailJobOnError(false);
jcsSession	com.redwood.scheduler.api.model.SchedulerSession	jcsSession.getUserName()
jcsJob	com.redwood.scheduler.api.model.Job	jcsJob.hold();
jcsOut	java.io.PrintWriter	jcsOut.println("This text will be printed to a specific output file on the job.");

Before Job Pre Running

Object	Class	Example Code
jcsOutLog	com.redwood.scheduler.infrastructure.logging.api.Logger	jcsOutLog.info("This is an informational message!")
jcsSession	com.redwood.scheduler.api.model.SchedulerSession	jcsSession.getQueueByName("UNIX_Generic");
jcsJob	com.redwood.scheduler.api.model.Job	jcsJob.hold();
jcsPreRunningContext	com.redwood.scheduler.api.scripting.variables.PreExecutingActionScriptObject	String strNewStatus = jcsPreRunningContext.getNewStatus();
jcsOut	java.io.PrintWriter	jcsOut.println("This text will be printed to a specific output file on the job.");

Before Job Post Running

Object	Class	Example Code
jcsOutLog	com.redwood.scheduler.infrastructure.logging.api.Logger	jcsOutLog.debug("This is a debug message!")
jcsPostRunningContext	com.redwood.scheduler.api.scripting.variables.PostRunningActionScriptObject	
jcsSession	com.redwood.scheduler.api.model.SchedulerSession	jcsSession.getSchedulerEntity()

jcsJob	com.redwood.scheduler.api.model.Job	jcsJob.restart();
jcsOut	java.io.PrintWriter	jcsOut.println("This text will be printed to a specific output file on the job.");

Values

Field	Description
Partition	The partition of the trigger, this also defines the scope of the trigger as it will only effect trigger-points of objects that are in the partition of the trigger.
Name	The name of the trigger
Application	The application the trigger resides in
Description	A description of the trigger
Comment	The comment for the trigger, helpful for other users to understand your trigger
Enabled	Only enabled triggers will have effect
Execution Order	The order at which the triggers are executed for one same trigger-point
Library	Your library of classes to use within the trigger
Log File Enabled	Should the trigger actions be logged? This is helpful for troubleshooting your trigger
Source	The source code of the trigger
Stub Source	The source stub, which contains the stub code with predefined objects

Example

This example trigger checks if job definitions or job chains have an application, and throws an exception when this is not the case. This trigger allows you to force users to put job definitions and job chains in applications.

See Trigger Examples for more example triggers.

```
import java.text.MessageFormat;
import com.redwood.scheduler.api.model.JobDefinition;
{
Object o = jcsTriggerContext.getSchedulerEntity();
if (o instanceof JobDefinition)
    {
      JobDefinition jobDef = ((JobDefinition) o);
      if ( jobDef.getParentApplication() == null)
          {
             String msg = jobDef.getName() +" has no application!";
             jcsOutLog.info(msg);
             throw new RuntimeException(msg);
          }
    }
}
```

See Also

- Trigger Examples
- Granting and Revoking System Privileges
- Granting or Revoking Object Privileges

Trigger Examples

Use

This topic provides examples of triggers to show you what can be done.

Tune Scheduling Parameters at Runtime

You can use a *Before Job Pre Running* trigger to change scheduling parameters.

Request 1

Jobs from many job definitions should run in a queue specified in a job parameter if the queue is open.

Solution 1

```
{
//get Job Parameter and Queue
JobParameter jp = jcsJob.getJobParameterByName("Queue");
Queue q = jcsSession.getQueueByName(jp.getInValueString());
if (jp != null && q.isOpen()) {
jcsOut.println("Setting job to run in Queue from JobParameter "+jp.getInValueString());
//Set Queue on the job
jcsJob.setQueue(q);
}
}
```

Alternate Request 1a

Jobs from a few job definitions should run in a queue specified in a job parameter if the queue is open.

Solution 1a

```
{
//get Job Parameter and Queue
JobParameter jp = jcsJob.getJobParameterByName("Queue");
Queue q = jcsSession.getQueueByName(jp.getInValueString());
if (jp != null && q.isOpen()) {
jcsOut.println("Setting job to run in Queue from JobParameter "+jp.getInValueString());
//Set Queue on the job
jcsJob.setQueue(q);
}
}
```

Overrule Errors

You use *Before Job Post Running* triggers to influence the final state of a job.

Request 2

A job reaches status *Error* because the exit code is higher than `0`. You want to override this, as exit codes `15` to `24` are in fact just warnings.

This should be done for most job definitions.

Solution 2

You specify a job parameter *ReturnCodeMapToWarning*, which contains a range, `15-24` in this case, on the job definitions for which you wish to override the final status.

You specify a *Before Job Post Running* trigger like the one below:

```
{
  JobDefinition jd = jcsJob.getJobDefinition();

  // Only run for user defined job definitions
  if (jd.getBehavior().longValue() == 0)
  {
    // Will contain ReturnCodeMapToWarning parameter if it exists
    JobParameter jp;
    // Will contain the range of codes to map to warning.
    String mapToWarning;
    // Is it in the range.
    boolean inRange;
    Long returnCode;

    // Look for the ReturnCodeMapToWarning parameter
    jp = jcsJob.getJobParameterByName("ReturnCodeMapToWarning");
    if (jp == null)
    {
    //Job parameter does not exist, do nothing
      mapToWarning = null;
    }
    else
    {
      mapToWarning = jp.getInValueString();
    }

    // Only do mapping if there is a sensible value
    if (mapToWarning != null && mapToWarning.length() >= 0)
    {
      returnCode = jcsJob.getReturnCode();
      if (returnCode == null)
      {
        // Handle null as 0
        returnCode = new Long(0);
      }

      // Use jcsSession.inRange to determine if the code is in range.
      if (jcsSession.inRange(returnCode.intValue(), mapToWarning))
      {
        JobNote jn = jcsJob.createJobNote();
        jn.setText("Setting status to Completed as "
                       + jcsJob.getReturnCode()
                       + " is in "
                       + mapToWarning);
        jcsPostRunningContext.setFinalStatus(JobStatus.Completed);
      }
    }
  }
}
```

Alternate Request 2a

Only a few job definitions are concerned.

Alternate Solution 2a

You specify a post-running action on the jobs concerned by this. Triggers fire for all jobs, if you specify the action on each job definition that is concerned by this, you can save processor cycles. Note that you will have to add the action to each job definition, though.

The code for the action is the same as for the trigger.

Change Status According to output

Request 3

Jobs reach status *Completed* because the return code is `0`, however, you want the job to reach status *Error* when an error was reported in the output of the job. A lot of job definitions are concerned by this issue.

Solution 3

You specify a job definition parameter containing the pattern to search for. The pattern should match the error as found in the output file. In this example, the error appears in the job file `stdout.log`

For example, an error in the job will be printed as follows:

```
myprog: cannot access /u01/backup/masalan: No such file or directory
```

In this case, the *ErrorSearch* job definition parameter could be set to `No such file or directory`, for example.

You use the *Before Job Post Running* trigger code below:

```
import com.redwood.scheduler.api.model.*;
import com.redwood.scheduler.api.model.enumeration.*;
import com.redwood.scheduler.api.search.*;

{
  JobParameter jp = jcsJob.getJobParameterByName("ErrorSearch");
  if (jp != null)
  {
    String errorSearch = jp.getInValueString();
    JobFile jf = jcsJob.getJobFileByName("stdout.log");
    SearchResultSet rs = jf.search(errorSearch);
    while (rs.next() && rs.getSearchResult().wasFound())
    {
      SearchResult sr = rs.getSearchResult();
      jcsOut.println("Found " + errorSearch + " at " + sr.getLineNumber());
      jcsOut.println(sr.getLineText());
      jcsPostRunningContext.setFinalStatus(JobStatus.Error);
    }
  }
}
```

Alternate Request 3a

Only a few job definitions are concerned.

Alternate Solution 3a

You specify a post-running action on the jobs concerned by this. Triggers fire for all jobs, if you specify the action on each job definition that is concerned by this, you can save processor cycles. Note that you will have to add the action to each job definition, though.

The code for the action is the same as for the trigger.

See Also

- Using Triggers

Using Actions

Use

Actions and triggers allow you to interact with user actions, the most common one being the final status of jobs. The main difference between triggers and actions is the scope, actions only apply to the definitions for which they are defined, triggers apply globally. With job definition actions, for example, you can change the status to `Completed` even if the job reached status *Error* based on advanced criteria. Actions are also available in the advanced Active Monitoring Module as well as in the import/export module, both require specific license keys that enable the respective module. The Active Monitoring Module requires the *Module.Alerting* and the import/export module requires the *Module.ImportExport* license key. Actions can make use of a library in which you can store frequently used classes.

Note

You must never persist a session in an action, that is done for you at a later point.

Note

Actions require RedwoodScript which in turn requires an Scripting license name *Module.Scripting*.

Import/Export Module

Actions are available in import rule sets to customize objects before they are written to the database. You can rename objects or assign them to an application, for example.

Import Rule Sets

The context of actions in import rule sets have a *SchedulerSession* object which allows you to rename the objects, for example, and the *jcsImportRuleSet* which is an iterator over the collection of objects that are to be imported. See the example below.

Object	Class	Example Code
jcsSession	com.redwood.scheduler.api.model.SchedulerSession	jcsSession.getQueueByName("PRD_QUEUE");
jcsImportRuleSet	com.redwood.scheduler.api.scripting.variables.ImportActionScriptObject	jcsImportRuleSet.getObjects();

Note

You should not persist the session in import rule set actions, this is done for you at a later stage.

Active Monitoring Module

Actions are available in the Active Monitoring Module in the following alerting objects:

- Alert Escalation – after the alert is escalated
- Alert Source – after the alert is created
- Email Alert Gateway – before the email is sent

Actions

The contexts of actions available in alerting objects have the Logger (jcsOutLog) as well as a object-specific context which is used to retrieve the *alert*, the *alertSource* or the *message*, for example.

Object	**Class**	**Example Code**
jcsOutLog	com.redwood.scheduler.infrastructure.logging.api.Logger	jcsOutLog.info("This is an informational message");
jcsAlertSourcePostAlertContext	com.redwood.scheduler.api.scripting.variables.AlertSourcePostAlertActionScriptObject	jcsAlertSourcePostAlertContext.getAlertSource();
jcsEmailAlertGatewayPreSendActionContext	com.redwood.scheduler.api.scripting.variables.EmailAlertGatewayPreSendActionScriptObject	jcsEmailAlertGatewayPreSendActionContext.getMessage();
jcsAlertEscalationPostAlertContext	com.redwood.scheduler.api.scripting.variables.AlertEscalationPostAlertActionScriptObject	jcsAlertEscalationPostAlertContext.isResend()

Example

Sample code that is to be used in *Import Rule Sets* and changes the application of all imported objects, except applications, to *Finance*, which has *Masalan* as parent application.

Checks are made for the imported object, making sure it can have an application (ApplicationObject) and that the imported object is not an application (to avoid errors, if the application or its parent are part of the import).

```
{
  Application parentApp = jcsSession.getApplicationByName("Masalan");
  Application app = jcsSession.getApplicationByName(parentApp, "Finance");
  for (final Iterator it = jcsImportRuleSet.getObjects().iterator(); it.hasNext();)
  {
    Object o = it.next();
    if (o instanceof ApplicationObject && !( o instanceof Application))
    {
      ((ApplicationObject) o).setParentApplication(app);
    }
  }
}
```

See Also

- Redwood Script
- Trigger Examples
- Triggers

Executing Your Own Classes from a Library

Use

The library allows you to store your RedwoodScript code for use in SAP CPS, in one or multiple triggers, for example, or in Redwood Expression Language.

You can write your own code directly in your library or add jar files to your library. RedwoodScript code in a library must always be in *Full* mode, that is start with `package com.redwood.scheduler.custom....` This code can then be accessed from triggers, actions, job definitions, period functions, and Redwood Expression Language.

Note

Custom library names must always start with `Custom_` or you will not be able to save the library.

Using Loggers in Libraries

A logger is an object that allows you to write log messages. Loggers have a category and a level. The category is the source of the message, and is arranged hierarchically. All loggers related to SAP CPS are in the category `com.redwood.`, under which there are many more sub-categories (and sub-sub-categories and so on). Loggers also have a level, this controls the amount of information that is logged from a particular source. The levels (in order of most to least important) are:

- FATAL - messages that will cause an immediate issue for the system or important customer functionality, that require human intervention to resolve.
- ERROR - errors that only effect part of the system (for example a job failing or a file not being present)
- WARN - potential errors or misconfiguration.
- INFO - informational messages, generally about progress in a long running task.
- DEBUG - detailed information about system progress.

The logger object has two methods for each level, one that takes a message, and another that takes a message and a throwable object (an exception).

```
Logger log = ...;

log.debug("Some information useful to developers is " + someVariableName);

log.info("Finished Pass 2/3");

try
{
  ...
}
```

```
catch (IOException ioe)
{
  log.error("Error accessing file", ioe);
  throw ioe;
}
```

Accessing Libraries in RedwoodScript

To use your methods from within RedwoodScript, you first need to set the library containing your code on the object, you can then use your code using one of the following:

- specifying `<package_name>.<class_name>.<method_name>`
- import your class using the following import statement `import <package_name>.<class_name>;` and use `<class_name>.<method_name>`

Note

You should not use the *jcs* prefix in the names of your methods, as this prefix is reserved for the system.

Accessing Libraries in Redwood Expression Language

You can use your methods in Redwood Expression Language throughout the product by specifying *REL Entry Points* with the following information:

- Name - the name of the method to be used in Redwood Expression Language
- FQ Class Name - the fully qualified class name
- Method Signature - the method syntax

Note

You should not mix code in a same class that is to be used in Redwood Expression Language and RedwoodScript. See Mixing REL and RS Code in a Library for more information.

Note

You should not use the *jcs* prefix in the names of your methods, as this prefix is reserved for the system.

Persisting Sessions in Libraries

It is not recommended to persist a session in your library code that you intend to use in Redwood Expression Language, as it can have side-effects. The behavior or side-effects might even change from one release to the next without prior notice.

Persisting a session in your library code used for RedwoodScript may have side effects as well, especially when the code is used in triggers, job actions, or import rule set actions. It is strongly recommended not to persist a session in your library code if it is used in triggers, job actions, or import rule set actions.

You may persist a session in your library code when it is used by alert escalations, alert sources, or email alert gateways and none of the previously mentioned objects.

Actions

Libraries support the following actions:

Action	Description
Import JAR	Import the code of a JAR file into the library
Export	Export the library into a CAR file
Edit	Edit the library
Edit Security	Edit the security of the library
Delete	Delete the library
Duplicate	Make a copy of the library to create a similar one
Expand All	Expand all libraries in the current filter
New	Create a new library
Filter > New Filter	Create a new library filter
Filter > Edit Filter	Edit current library filter
Filter > Delete	Delete current library filter

Finding Libraries

You can search for libraries using filters and the *Search Libraries* box on the Libraries tab. This box is known as the *intelliSearch* box and located under your username on the top right-hand side of the user interface. Filters allow you to specify a list of objects with static criteria. IntelliSearch allows you to specify complex queries in a simple way using prefixes. Prefixes are used to specify which property you are searching in and have short and long syntaxes. For example, if you want to display all libraries with the term *trigger* in the comment, you would use the search criteria as follows:

```
c:trigger
```

You can search more than one property, as follows:

```
c:trigger n:JV
```

Note

No spaces should be entered before or after the colon (*:*).

See the Advanced Object Search for more information.

The following table illustrates the available prefixes for libraries:

Prefixes	Description
n, name	searches the name property
c, com, comment	searches the comment property
d, desc, description	searches the description property
a, app, application	searches the application property

Deleting Libraries

You can only delete libraries when no other objects relate to them. For example, if there are job definitions that use the library, the library cannot be deleted until all job definitions that use it have been modified. You can see all job definitions that relate to the library in *Related Objects* in the lower detail pane and on the show page.

The table in related objects contains three columns:

- Type - the type of object with a link to it
- Related Object - the name of the object with a link to it
- Used As - objects can sometimes be used in different roles

Security

Privilege	**Description**
Library.Create	Create libraries
Library.Delete	Delete libraries
Library.Edit	Edit libraries
Library.View	Access libraries

You can grant privileges on two levels, *Access* and *Admin;* a privilege granted on *Admin* level allows the grantee to grant the privilege to other users. These privileges can be granted *system-wide*, per *partition* or *isolation group*.

If you have the security module, which requires the *Module.Security* license key, you have an additional *Security* tab on the library. It allows you to specify which users can access, edit, and delete the library.

- Granting and Revoking System Privileges
- Granting or Revoking Object Privileges

Procedure

Create you own package in the custom library

1. Navigate to *Scripting* > *Libraries*.
2. Choose *Edit* from the context menu of the library.
3. Choose the *Source* tab, choose *New* and enter your code.
4. Choose *Save* to save your newly added package.
5. Choose the *REL Entry Points* tab and fill in the following information:
   a. *Name* - the desired name of the method, the name should remind others of what the method does.
   b. *FQ Class Name* - the fully qualified name of the class, <package_name>.<class_name>.
   c. *Method Signature* - the signature of the method, e.g. its name in your code and the types of arguments that should be provided.

Add a jar file to the library

1. Navigate to *Scripting* > *Libraries*.
2. Choose *Add Jar* from the context menu of the library.
3. In the *Import Jar File Into Library* make sure the *Library Name* is correct and navigate to the jar file. The jar file needs to be accessible to the system where SAP CPS is running.

4. Choose *Send* and wait until the jar file has been loaded. The *System_Import_JarFile* job definition gets submitted automatically to upload the jar file, if this job fails, the jar file has not been added. Please monitor the job and inspect the *stderr* and stdout files for more information.

Example

In this example, one class is used for both RedwoodScript and Redwood Expression Language. This is for illustration purposes only to show you how to access your methods from both and should not be done in a production environment. Always keep methods for RedwoodScript and Redwood Expression Language in different classes.

1. Navigate to *Scripting* > *Libraries*.

2. Choose *Edit* from the context menu of the *Custom* library.

3. Choose the *Source* tab, choose *New* and enter the code under the *Example Code* section.

4. Choose *Save* and choose the *REL Entry Points* tab and fill in the information under the *REL Entry Points* section:

Example Code

```
// Example code to illustrate
package masalanCustomString;

public class concat
{
public static String concatString(String str1, String str2)
   {
       if (str1 == null && str2==null ) return "null";
       return str1.concat(str2);
   }
}
```

REL Entry Points

- *Name* - `concatenate`
- *FQ Class Name* - `masalanCustomString.concat`
- *Method Signature* - `concatString(String,String)`

Use in Redwood Expression Language

In a job parameter:

```
=Custom.concatenate('Masalan ', 'Inc.')
```

You can also concatenate strings from parameters. If you do not make it a runtime parameter, however, it will only concatenate the default values of the parameters and not the values entered at submit time.

The following example was used in a job definition with three parameters, the first two had a *Display Order* of `0` and `1`, respectively the third had a *Display order* of `99`. The display order attribute defines the order at which parameters are displayed in the submit wizard as well as the evaluation sequence.

```
=Custom.concatenate(parameters.param1, parameters.param2)
```

Use in RedwoodScript

In a job definition

1. Create a RedwoodScript job definition, choose *RedwoodScript* as job definition type and the library you added the above code to as library.

2. Give the job definition a name on the *Job Definition* tab.

3. On the source tab enter the following code:

```
{
jcsOut.println(masalanCustomString.concat.concatString(Param1, Param2));
}
```

An alternative, is to import the class, so you do not have to specify the package name every time:

```
import masalanCustomString.concat;
{
jcsOut.println(concat.concatString(Param1, Param2));
}
```

Mixing REL and RS in a Library

Use

You should not mix code for REL and code with constructors in any one class, as this can have side-effects. You encounter the issue when constructors that take arguments are used.

Example

The following examples illustrate what should and should *not* be done:

The following class can be used with REL.

```
public class MyClass
{
  public String myFunction(String argument)
  {
    // Code
  }
}
```

The following class can be used with REL.

```
public class MyClass
{
  private int someVariable;

  public MyClass()
  {
    someVariable = 0; // Default value
  }

  public MyClass(int newValue)
  {
    someVariable = newValue;
  }

  public String myFunction(String argument)
  {
    // Code
  }
}
```

The following example will have side-effects

```
public class MyClass
{
  private int someVariable;

  public MyClass(int newValue)
  {
    someVariable = newValue;
  }

  public String myFunction(String argument)
  {
    // Code
  }
}
```

Chapter 9. Monitoring Automated Processes

Monitors give you an overview of all running jobs, all alerts, messages and the status of events.

The *Jobs* monitor is the main dialog for displaying and interacting with jobs. The *Calendar* monitor allows you to visualize the jobs in a calendar view and helps you organize your processes.

1. Monitoring and Managing Jobs
   a. Job States
   b. Filtering Jobs and Events
   c. Monitoring Jobs in a Calendar view
2. Monitoring Events
3. Interacting with Operator Messages
4. Monitoring System Performance
   a. Showing Monitors as Graphs in the Dashboard
5. Monitoring SAP Systems
6. Configuring Alerting
   a. Triggering Alerts
      i. Job Alerts
      ii. Process Server Alerts
      iii. Custom Alerts
   b. Defining Escalation Routes
   c. Configuring Email Alert Gateways
7. Interacting with Operator Messages
   a. Receiving Operator Messages on your Desktop
   b. Creating Advanced Reply Expressions
8. Integrating SAP CPS in your SNMP Landscape

Monitoring and Managing Jobs

Use

The job monitor is part of the Monitoring section in the navigation bar and gives a central overview of all jobs and their statuses on all managed servers in any queue. From the job monitor jobs can be monitored as they execute and you can perform a number of actions on the jobs. The job monitor displays a number of properties of jobs in columns, which can be customized, to give you an up-to-date overview of the jobs and their states.

The job monitor consists of two panes:

- The job monitor pane displays a hierarchy of all jobs that match a chosen filter.
- The job preview pane displays extensive run-time data of the currently selected job.

The preview pane will show job details. A diagram similar to the job chain edit diagram is available from the context menu for job chains. To view the details of a job chain or call, choose the job id.

Note

On clusters with multiple nodes, job id's are not necessarily sequential in time. Each node in the cluster will have arrange of job id's.

Editing Jobs with a Recurrence

When you edit a job, you can change the scheduled start time, queue or other properties of a single job, as long as it has not reached the *Running* status. This is done, using the actions under *Scheduling*.

You can also edit the properties for jobs of a recurrence; this is done using the actions under *Recurrence*. Changes performed on a job in the recurrence, regardless of its status, will generally have effect on all jobs of the recurrence that have not reached the *Running* state at the time of the change. If you want control over which jobs are affected, put the job on *Held* to make sure you have enough time to perform the change. Once you have finished adapting the recurrence, do not forget to release all jobs you held.

Actions

Jobs support the basic actions of Help, Refresh, Close tab, and Close all other tabs.

Jobs have additional actions:

Action	**Description**
Remove Old Jobs	Remove old jobs that have reached a final status
Add Note	Add a note to the job
Explain Job Status	Explain the current job status, only works if the current job status is *Queued*
Recurrence > Delete Recurrence	Delete the jobs of the job group that have reached a final status.
Recurrence > Edit Recurrence	Edit scheduling data of the job group.
Recurrence > Hold Recurrence	Pause the recurrence; this can be performed on any job of the job group regardless of its status.
Recurrence > Release Recurrence	Resumes a paused recurrence.
Recurrence > Cancel Recurrence	End a recurrence.
Recurrence > Restart Recurrence	Create a new job group with the recurrence.
Scheduling > Edit	Edit scheduling data of the job. To edit a job, the job must be in an initial state, not *Running*.
Scheduling > Resubmit	submit a new job based on this job (same definition, parameters, and so on). This will pop up the new window to edit the values before the job is submitted.

Scheduling > Hold	this will hold the job, preventing it from being executed until it is manually released.
Scheduling > Release	this will release the job, allowing it to be executed automatically when all its scheduling constraints are met.
Scheduling > Enable	this enables a previously disabled job chain step or call.
Scheduling > Disable	this disables a previously enabled job chain step or call.
Scheduling > Kill	kill a running job by sending it a signal to stop running. Killing a job may not always be instantaneous.
Scheduling > Cancel	cancel a job that has not yet started running. This will prevent the job from ever starting.
Scheduling > Restart	submit a new job based on this job (same definition, parameters, and so on). This will submit the job immediately, without prompting.
Scheduling > Cancel recurrence	Cancel all recurrences of the job (if the job is submitted with a recurrence).
Show diagram	Show the job in a diagram, useful for job chains and SAP process chains.
Delete	Delete the job, it has to be in a final state. If its state is pre-running, you must cancel it first. If the job is running, you must kill it before you can delete it.
New Alert Rule	Allows you to create a new alert rule for the job with a default *Name* and *Name Pattern*.
Get Support Files	Allows you to retrieve the log files that might help you troubleshoot a problem. Only available for jobs that have not reached status *Completed*.
Filter > Create Filter	Create a job filter
Filter > Edit Filter	Edit the current job filter
Filter > Delete Filter	Delete the current job filter

The detail pane shows the different values listed in the *Values* section below. Among these are *Job files*, which can be viewed, downloaded, and deleted once the job has completed.

Procedure

To view parameters of a job:

1. Choose *Monitoring > Jobs*.

2. Choose the *Refresh* to update existing jobs and show new jobs.

3. Choose an auto-refresh interval.

4. Expand a job chain, notice the child steps.

5. Expand a step.

6. Select a job, notice that the detail pane appears.

7. Scroll down and expand the *Parameters* group.

8. The overview displays the value when the job started (In Value) and the value once it finished (Out Value) of every parameter (if applicable).

9. Expand the *Files* node. These are the output files of the job.

10. Choose a file to view its contents.

Values

More information about the available columns in the job monitor can be found in the Available Columns in the Job Monitor section of the documentation.

See Also

- Advanced Job Search topic
- Viewing Jobs in a Calendar View
- SAP CPS Job Processing
- Job States
- Available Columns in the Job Monitor

Available Columns in the Job Monitor

Use

The following columns are available in the job monitor:

Field	**Description**
Description	The name of the job. Equals the description of the submitted job definition that created this job.
Job Id	The job id for this Job. This is unique for every Job.
Job Definition	The job definition that was submitted to create this job.
Status	The current status of the job.
Job Files	The output files of the job. Choose the links in the table to display the content of the output files.
Queue	The queue the job was submitted into.
Submit Frame	The submit frame the job was submitted into.
Process Server	The process server assigned to handle the job.
Priority	The priority of a job in the range 1 to 100. Higher values have precedence.
Run Start	The date and time the job status started executing.
Run End	The date and time when the job finished.
Scheduled Start Time	The scheduled start date and time of the job. This is calculated by the dispatcher, and is dependent on any time windows associated with the Job, or the jobs queue, and the requested start time set by the user. This value is also set on a best effort basis when this job was submitted.
Creation Time	The creation date of this Job.
Last Modification Time	The last modification date of the job.
Next Submit Base Time	The time that will be used as a start time for the calculation of the next requested start time for a job with a submit frame, or for the requested start time of a new job where the job has an infinite recurrence pattern.

Requested Start Time Type	Has the requested start time been explicitly set by the user, or has it been inherited from the Job's parent.
Requested Start Time	This is the time that the user has requested that the job start. In the absence of other restrictions, for example, time windows, this will be the scheduled start time.
Restart Count	The number of restarts left for this job.
RemoteCreationTime	The creation time of the job in the remote system
RemoteEvent	The remote event
RemoteId	The remote unique identifier
RemoteRunEnd	The remote end time of the job
RemoteRunStart	The remote start time of the job
RemoteStatus	The remote status
RemoteSystem	The remote system
Job Parameters	The parameters for the job.
Job Wait Events	Events that need to be raised before the job runs.
Job Raise Events	Events that will be raised when the job reaches a certain state.
Job Messages	Messages generated by the system and the jobs specific for this job.
Job Notes	Ad-hoc job notes written by end-users attached to this job.

Note

On clusters with multiple nodes, job id's are not necessarily sequential in time. Each node in the cluster will have arrange of job id's.

See Also

- Monitoring and Managing Jobs
- Advanced Job Search topic
- Viewing Jobs in a Calendar View
- SAP CPS Job Processing
- Job States

Job States

Use

Job states are there to show which stage a job has reached. There are four basic stages in which one can classify the various job states.

1. initial
2. pre-running
3. running
4. final

Every job goes completed job went through all four stages, however, a job cannot "revert" back to a previous stage. For example, a *Running* job cannot return to a pre-running stage status.

Initial Job States

Status	Description
New	The process has been submitted, but not yet validated.
Chained	A job chain is requested and all jobs in the chain have been created and are awaiting execution
Disabled	The process server will not execute the job chain job due to a manual request of the user; when the job chain is restarted, job chain jobs that have the status *Disabled* will not be executed.

Pre-Running States

Status	Description
Scheduled	The process has a valid start time and is waiting for execution
QueueHeld	The processes queue has been manually paused
Held	The execution of the process has been manually paused.
EventWait	The process is waiting for an event to be raised.
LockWait	The process cannot obtain a process lock and is waiting for one to be released.
Queued	The start time is known, but the process cannot (yet) be executed.
Dispatched	The process has been dispatched to a Service, but the Service has not yet started the process.
Assigned	The process has been accepted by the Service, but it has not been started yet.

Running States

Status	Description
Running	The process is being executed by the Service. Once it has entered this state, the JobService that is running this job no longer needs any access to the JobParameters of this Job.
Waiting	The process is waiting for the result of the execution of another process.
Killing	A request was made to kill the process and the JobService has honored that request.
Canceling	A request was made to cancel the process and the JobService has honored that request.
Console	The process has issued an OperatorMessage to the Messages monitor and is waiting for a reply.

Final States

Status	Description
Completed	Process execution finished successfully.
Error	The execution of the process failed.
Killed	The process was terminated while it was running (process status was *Running*, *Waiting* or *Console*).
Canceled	The process was terminated before execution commenced.

Skipped	The process server has not executed the job chain job due to a pre-condition or the usage of the "Start at specified step" scheduling parameter. When the job chain is restarted, the process server will re-evaluate job chain jobs with the status SKIPPED.
Ignored	This process was ignored as a result of being Disabled when the job chain was submitted.
Unknown	The Service was terminated while the process was running (process status was RUNNING, WAITING or CONSOLE).

Special States

Status	**Description**
Modified	The parameter list of a job in a job chain have been modified - the process server cannot proceed with the job execution because not enough parameter values are available - operator intervention is required.
Never	Due to conflicting time windows, time window elements or a shutdown process server, the process shall not execute for at least two years. The start time is set to December 31st 2999
Overdue	The process did not run in the first time window opened after the process start time

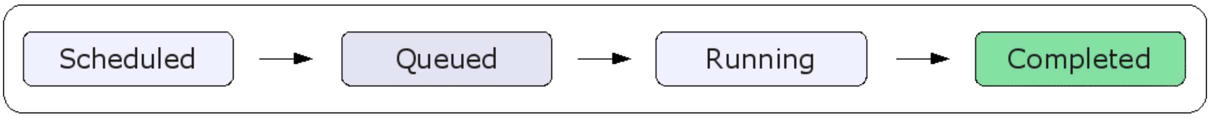
Note

It can occur that the job gets deleted in the SAP System before the final status has been retrieved by SAP CPS. In this case, the job will get status *Unknown* in the job monitor.

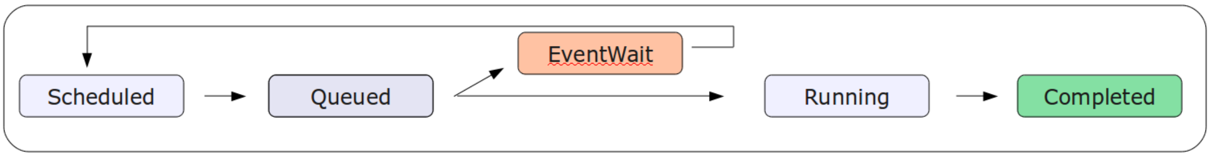
Diagrams

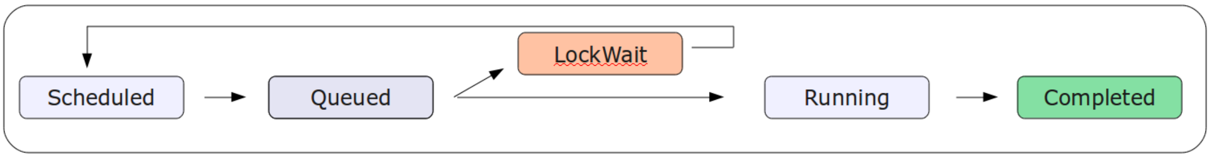
Understanding Status changes

A simple job would pass the stages below, in fact, not all states are displayed here, most of the time you will not see jobs in status *New*, *Dispatched*, or *Assigned* as these states are transitional and are immediately reacted upon:

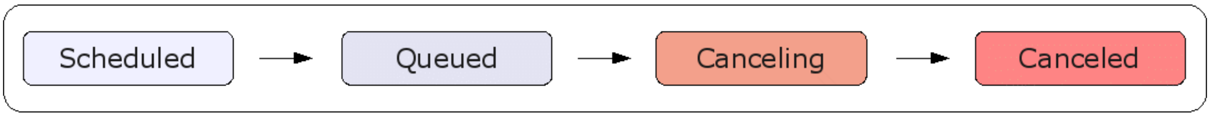


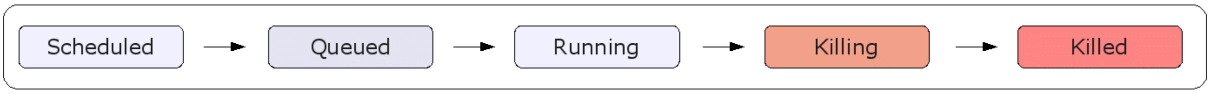
Wait events and locks force the job to wait until all wait events are raised or until the lock is released:





Jobs that have not yet reached the *Running* status can be canceled; jobs that have reached the *Running* status and not yet reached a final status must be killed if you want to interrupt them. When you cancel or kill a job, the job will remain in one of the two transition states *Canceling* or *Killing* for a short time, depending on the system load, before it reaches its final status.





Job chains

A submitted job chain consists of the following jobs:

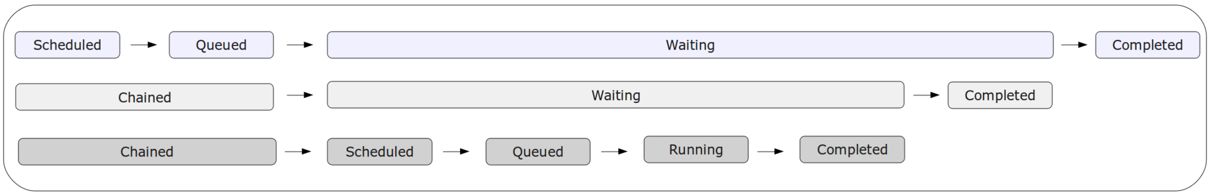
- a job for the chain itself
- a job per step
- a job per job chain job

A basic job chain with one step and one job chain job that got submitted has three jobs.

When a job chain is submitted, job chain steps and job chain jobs will remain in status *Chained* or *Disabled* until they are started. *Disabled* Jobs immediately reach status *Ignored* when the process server processes them. *Disabled* and *Ignored* steps have been disabled by an operator.

You can skip jobs by submitting a job chain at a specific step, all jobs that would have been executed before the step in question will be skipped. Precondition functions can be used to have specific jobs run at specific times only, for example, a job chain that runs every day could have one or more jobs with a precondition restricting these jobs to run only on Mondays. The job chain would run every day, the jobs that were to run on Mondays will be *Skipped* on other weekdays.

By default, a job chain job will wait until all its child-steps have completed successfully or a child-step completed unsuccessfully. Steps will wait in the same way for their job chain jobs. When a job in a step fails, the step waits for the other jobs in the step to reach a final state and is set to *Error*. At this moment the job chain will be set to *Error* and steps that came after the failed step are not executed and remain *Chained*. You can use *Status Handlers* to customize this behavior.



By default, when you kill or cancel a job chain step or a job chain job, the main job chain job will be set to status *Error*. You can use *Status Handlers* to customize this behavior.

The final status of a job chain is determined by the status of the last step that reached a final state; the following final states are special: `Disabled`, `Skipped`, `Ignored`. This means that if you have a job chain of which the last step is `Disabled`, for example, the final status of the chain will be *Completed*.

The status of the job chain to also be set to reflect the status of the last executed step. A registry key and comment keywords have been introduced to allow you to toggle the behavior. The keyword in the Comment of the job chain definition has precedence over the registry key.

The following comments can be used to enable the new behavior:

```
@useLastExecutedStepStatus=true
```

To disable, use the following:

```
@useLastExecutedStepStatus=false
```

Just place the keyword (including @ and the desired value, true or false) into the *Comment* field on the *Comment* tab of a job chain.

The registry key that governs the default behavior:

```
/configuration/jcs/jobchain/useLastExecutedStepStatus
```

The registry key should be set to either `true` or `false`.

See Also

- Monitoring and Managing Jobs
- Advanced Job Search topic
- Viewing Jobs in a Calendar View
- SAP CPS Job Processing
- Available Columns in the Job Monitor

Filtering Jobs

Use

In large organizations, with many jobs, it is common to have operators monitor a subset of these jobs. To allow the user to just see the objects he is interested in, SAP CPS features filters and intelliSearch.

Default filters are already available:

For the job monitor:

- All (non-maintenance) jobs
- For next 24 hours

You can also define own filters using a subset of the conditions listed below. Each condition can have one or more values. For each value entered, conditions can be used to show or hide specific jobs.

The following conditions can be used to create customized job filters:

- Where the job has no parent
- Where the job is a background job
- Where the job description contains specific keywords
- Where the job is on a process server
- Where the job is in a queue
- Where the jobs remote system name contains specific keywords

- Where the job is scheduled to start after X days ago.
- Where the job run date is after X days ago.
- Where the job end date is after X days ago.
- Where the job has status X
- Where the job is from a named job definition
- Where the job is from a job definition whose name starts with a prefix
- Where the job is from a job definition whose name contains specific keywords
- Where the job belongs to an application
- Where the job priority is X
- Where the job is scheduled to start within the last X days or within the next Y days
- Where the job is scheduled to start after date
- Where the job is owned by user/role
- Extend filter with user defined SQL

You can use the ANSI SQL92 standard `where-clause` syntax, for user defined SQL, you have to omit the `where` keyword. See the Job Filter Columns section for a list of columns.

Using User Defined SQL to display all jobs with a job id above `12345` except background jobs

```
JobId > 12345 and Behavior <> 2
```

The `Behavior` column can be used to differentiate *background jobs* from others, background jobs have a behavior set to `2`. You can combine the different preset and user defined clauses, so *Where the job is a background job* on the *Include* tab is equivalent to `Behavior = 2`.

The above example using the *Where the job is a background job* on the *Include* tab would only require the following:

[x] Where the job is a background job

```
JobId > 12345
```

Procedure

Defining a job filter

1. Navigate to *Monitoring* > *Job Monitor*
2. Choose *Filter* > *New Filter* from the context menu.
3. Specify a name, a description (display name), and a number of conditions that are to be met.
4. For each condition, specify one or more values by choosing the blue text
   a. In the new window you must choose *Add* for each value of a condition.

b. Choose *OK* to close the window.

5. Choose *Save and Close*.

Advanced Job Search

Use

To allow you to search overviews quickly, a job search box, named intelliSearch, has been implemented. This search is able to use the following phrases as criteria:

- numbers - job ids
- times - of *requested start*, *run start,* and *run end* time
- remote - remote information
- names - names of queues, process servers, job descriptions or job definitions

Prefixes are available to restrict the amount of results, all searches are case-insensitive.

Using Numbers

Any number is assumed to be a job id.

For example, `111` would display the job with job id `111`. If this job is part of a job chain, the job will be shown expanded in the containing job chain. When searching on job id, the job(s) will also always be selected.

Times

Searches for all jobs between a specified time. The search-criteria has to be specified in 24h times. If no prefix is used, the search is done on all three times:

- *requested start time*
- *run start*
- *run end*

There are different ways of searching by time:

The time is specified using the following codes: seconds (s), minutes (m), hours (h), weeks (w), months (M), and years (y).

- *hh:mm* - without a range, the default range is +/- 15 minutes.
- *hh:mm:ss* - without a range, the default range is +/- 1 minute.
- *hh:mm-hh:mm* or *hh:mm:ss-hh:mm:ss* - range is from start time to end time.
- *hh:mm~nn{s,m,h,d,w,M,y}* - the number indiated by `nn` defines a range.
- *hh:mm~now* or *hh:mm~* - if you use `now` in the range, or leave one blank, the current time is used.

For example, `22:00-23:00` would show all jobs that have one of the three times in the range of *22:00-23:00* and `22:00~90m` would result in all jobs that have one of the three times in the range of 20:30-23:30 to get listed.

When you want to specificy a specific time, use a prefix.

Prefixes

- *q, requested* - Requested start time
- *s, runstart* - Run Start time
- *e, runend* - Run End time

For example, `runend:22:00-23:00` would restrict the above example to show only jobs which have a *Run End Time* in the specified range and `q:22:00~90m` would restrict the above example to display jobs which have a *Requested Start Time* in the specified range.

Remote

Allows you to query for jobs that run on remote systems with information of the remote system.

The following prefixes are available:

- *rid, remoteid* - the remote id.
- *rsys, remotesystem* - the remote system
- *rstat, remotestatus* - the remote status of the job

For example, `rid:A12S34H3` would display the job which has the remote id *A12S34H3* and `remotestatus:error` would display all jobs that have the remote status *error*.

Names

When a string is entered without a prefix, the search is done in the following order.

If the search string matches

1. a queue name, return the results for queue matches
2. an SAP queue name, return the results for SAP queue matches
3. an SAP process server, return the results for SAP process server matches
4. a process server name, return the results for process server matches
5. a job definition name, return the results for job definition, matches
6. a application name, return the results for application matches search for a partial or complete match of the job name (description).

Prefix

The following prefixes are available

If you want to search for a specific name, you can use one of the following prefixes.

- q, queue - find jobs from a queue
- sapq - find jobs that run in an SAP System queue

- sapps - find jobs that run on an SAP System process server
- ps, processserver - find jobs that run on a process server
- jd, jobdefinition - find jobs from a job definition
- a, application - find jobs from an application
- j, jobname - find jobs with a job name
- bg - include background jobs, default is *false*

For example, `q:System` would display all jobs are in the queue named *System* and `jobdefinition:System_Info` would display all jobs of the job definition *System_info*.

You can also specify multiple combinations of the above, when you specify more than one queue, for example, jobs of all specified queues will be displayed. For example, `PR1_Queue PR2_Queue PR3_Queue q:QA4_Queue` would display all jobs from the four queues *PR1_Queue*, *PR2_Queue*, *PR3_Queue*, and *QA4_Queue*. `q:PR1_Queue PR2_Queue PR3_Queue ps:PR1_ProcSerVer` would for example display all jobs that are on the specified process server as well as in the specified queues.

Note

Job descriptions can contain spaces, to specify a name with spaces, use double quotes (").

Note

A search on jobid, jobdefinition or jobname does not need a `bg:true`, because these automatically include all jobs that match the search criteria.

Note

If a specific search returns no results, the overview will be empty.

See Also

- Monitoring and Managing Jobs
- Viewing Jobs in a Calendar View
- SAP CPS Job Processing
- Job States
- Available Columns in the Job Monitor

Viewing Jobs in a Calendar View

Use

To allow you to view all your currently planned processes in a Calendar, switch to the *Calendar* monitor. You use the *Calendar* monitor to organize your processes in a timescale.

Your processes are listed as lines or boxes, when the jobs run for a long time. Expand them to view information about the process, like the time it is scheduled for and the description of the job.

The following color-codes are used to indicate the status of the job:

- *green* - the job has reached the status *Running* or *Completed*
- *red* - the job has reached a final status among *Error, Killed or Canceled*
- *purple* - the job has not reached a final status; the job might be *Scheduled*, *Queued*, waiting for locks or events.

The calendar is shown in a weekly view, to see jobs in previous or future weeks, use the arrows next to the refresh icon; you can also choose *Previous* or *Next* from the context menu.

See Also

- Monitoring and Managing Jobs
- Advanced Job Search topic
- SAP CPS Job Processing
- Job States
- Available Columns in the Job Monitor

SAP CPS job Processing

Use

The life-cycle of a job starts when the job is created and ends when the job has reached a final status. The job goes through various stages, each illustrated by a job status. The process that controls jobs is named the dispatcher. The dispatcher assigns jobs to process servers and queues for execution and is responsible for the life-cycle of the job.

The order in which the dispatcher stars jobs is defined as follows:

1. Based on priority of the job, (highest first)
2. Based on the requested start time of the top level parent of the job, (oldest first)
3. Based on the requested start time of the job, (oldest first)
4. Based on the job id, (smallest first)

Third-Party Systems

Some jobs represent workload performed in external third-party systems; for example an ERP system. In this case, the job start time in SAP CPS is the time at which the process server sends the request to the external system. So third-party systems can have a very slight delay in actual processing, which can be seen in the discrepancy between start time in SAP CPS and start time in the remote system. The discrepancy depends on the on the resource utilization on both the SAP CPS server and the remote system as well as network connectivity. Other factors might also have an impact, depending on the architecture of your data-center.

The process server communicates the status changes in the remote system back to SAP CPS; these are available in the *Job Monitor*.

See Also

- Monitoring and Managing Jobs

- Advanced Job Search topic
- Viewing Jobs in a Calendar View
- Job States
- Available Columns in the Job Monitor

Monitoring Events

Use

The event monitor is part of the Monitoring section in the navigation bar and gives an overview of all events and their statuses. From the event monitor the status of events can be monitored, see which and how many events are pending to be raised and clear raised events.

The event monitor consists of two panes:

1. The event monitor pane displays a list of all events and their status that match a chosen filter.
2. The event detail pane displays run-time data of the currently selected event and the ability to delete and clear events.

The possible event states are:

- *Pending* - The event has been created, by raising the event definition, but has not yet been raised by the event processing engine, either because it has not been processed, or because there is already another event for the event definition that has been raised.
- *PendingRaised* - The event definition is in the process of being raised by the event engine.
- *Raised* - The event has been raised, and has marked the corresponding event definition as having been raised. There can be at most one raised event per event definition at any point in time.
- *Cleared* - The event was raised in the past, but has now been cleared, and is being kept for a historical context, or has been passed to a job that was waiting on the event.

The following filters are built-in:

- All events
- Pending
- Raised
- Cleared
- Pending and Raised
- Raised and Cleared

Actions

Events support the following actions:

Action	Description

Clear	Clear the event
Delete	Delete the event
Expand All	Expand all events in the current filter
Filter > New Filter	Create a new event filter
Filter > Edit Filter	Edit current event filter
Filter > Delete	Delete current event filter

Procedure

To monitor events:

1. Choose *Monitoring > Events*.
2. Choose the *Refresh* button to update existing events and show new events.
3. Choose the event in the monitor to show the event's definition in the detail pane.
4. From the context menu, choose *Clear*.
5. From the context menu, choose *Delete*.

Values

Field	**Description**
Event Definition	The event definition that was raised to create this event.
Raised Sequence	Unique id for this event generated when it was raised. This is only guaranteed to be unique within each event definition, that is, if two event definitions are raised, they may have the same raised sequence, however, if the same event definition is raised twice, then each raise is guaranteed to generate an unique raised sequence.
Status	The current status of the event.
Raiser Comment	A comment made by the raiser when it raised this event.
Job Raise Events	Events that will be raised when the job reaches a certain state.
Job Wait Events	Events that need to be raised before the job runs.
Creation Time	Timestamp that notes when the event definition was created.
Last Modification Time	Timestamp that notes when the event definition was last modified.
Raise Time	Timestamp that notes when the event definition was raised.
Related Objects	Lists related objects such as jobs that waited on or raised the event.

Filtering Events

Use

In large organizations, with many events, it is common to have operators monitor a subset of these events. To allow the user to just see the objects he is interested in, SAP CPS features filters and intelliSearch. Default filters are already available:

- All events

- Raised and Cleared

You can also define his own filters using a subset of the conditions listed below. Each condition can have one or more values. For each value entered conditions can be used to show or hide specific events.

The following conditions can be used to create customized event filters:

- Where the event has status X
- Where the event name starts with a prefix
- Where the event raise date is after X days ago.
- Where the event raise date is after date
- Where the event was raised by job definition
- Where the event was waited for by job definition
- Where the event definition belongs to an application
- Where the event was modified manually
- Extend filter with user defined SQL
- Where the Event is a built-in one

You can use the ANSI SQL92 standard `where-clause` syntax, for user defined SQL.The available columns are listed in the Event Filter Columns section.

Example

Using user defined SQL to display events which were raised by a subject with id `82171`.

```
RaiserUniqueId = 82171
```

Procedure

Defining an even filter

1. Navigate to *Monitoring* > *Event Monitor*
2. Choose *Filter* > *New Filter* from the context menu.
3. Specify a name, a description (display name), and a number of conditions that are to be met.
4. For each condition, specify one or more values by choosing the blue text
    a. In the new window you must choose *Add* for each value of a condition.
    b. Choose *OK* to close the window.
5. Choose *Save and Close*.

Event Filter Columns

Use

The following columns are available for the use in event filters for user defined SQL:

Columns	Description
ClearerComment	Comment of the clearer
ClearerObjectDefinition	Object definition of the clearer
ClearerUniqueId	Unique Id of the clearer
ClearTime	Time the event was cleared
CreationTime	Creation time of the event
CreationTimeZone	Time zone at creation
EventDefinition	Event definition the event is based on
LastModificationTime	Last modification time
LastModifierSubject	Last modifier
RaisedSequence	Raised sequence number of the event
RaiserComment	The comment of the raiser
RaiserObjectDefinition	The object definition that raised the event
RaiserUniqueId	Unique Id of the raiser
RaiseTime	Time the event was raised
Status	Current status of the event
UniqueId	Unique Id of the event

Monitoring System Performance

Use

The performance monitor is part of the *Monitoring* section in the navigation bar.

The monitor displays an hierarchical overview of all monitors, and their current severity. The severity is propagated upwards through the hierarchy so a parent monitor shows the highest severity of its children.

The Monitor consists of two panes:

1. The monitor pane displays a hierarchy of all monitors.
2. The detail pane displays run-time data of the currently selected monitor.

Three types of monitors are shown:

1. Nodes - A monitor that can contain sub-monitors.
2. Checks - A monitor that defines conditions upon which events are to be raised.
3. Links - A monitor that links to another monitor (node or check).

SAP System Monitors

If you have connected SAP CPS to SAP Systems, then the default monitor-tree contains monitors for the SAP Application Servers. To import the bare monitor tree from transaction **RZ20**, you need to fill in the *XAL* tab on the SAP System object and import the monitors. To import individual CCMS monitors, that you access via transaction **RZ20** in SAP GUI, you need to submit the *SAP_ImportCcmsMonitors* job definition.

Severity

The severity gives you information about the extent of the issue and is indicated by a number ranging from `0` to `100`. SAP Monitors that are not monitored by SAP CPS have a special severity of `-1`, which means that you have to import the monitor with the job definition*SAP_ImportCcmsMonitors* if you want to use it.

The primary function of the monitor is to check no monitors have a severity above `0`. You can create Dashboard dials from within the monitor to get a graphical representation of the monitor values.

Actions

Monitor Nodes support the following actions:

Action	**Description**
Export	Export the monitor node
Edit Security	Edit Security of the monitor node
Expand All	Expand all monitor nodes

Procedure

To monitor components perform the following tasks:

1. Navigate to *Monitoring* > *Monitors* to get an overview of all monitors.
2. Choose the *Refresh* button to update the monitor.
3. Choose the [+] icon to expand the monitor nodes and links and show the child monitors. Choose the [-] icon to collapse and hide the child monitors.
4. Choose a monitor to show the monitor details in the lower detail pane.

To create a dashboard dial:

1. Navigate to *Monitoring* > *Dashboard*.
2. If you do not have a dashboard, you need to create one:
    a. Choose *Create* from the context menu, fill in a name and choose *Save & Close*.
3. Select your dashboard in the drop-down menu.
4. Navigate to *Monitoring* > *Monitors*.
5. Expand the monitor-tree until you have located the leaf-node you would like to use for the dial.
6. Choose *Create Dial* from the context menu.

Values

Field	Description
Parent Monitor Node	The parent monitor node this monitor belongs to.
Name	The unique name of this monitor.
Description	The description of the monitor.
Comment	A comment about the monitor.
Actual Severity (Node and Check monitors only)	The current severity of the monitor, or -1 if it has not been set before.
Remote Id (Node and Check monitors only)	Unique identification of the monitor in the remote system.
Linked Monitor (Link monitors only)	The linked monitor (Node or Check monitors).
Monitor Events (Node and Check monitors only)	Event to be raised when a monitor reaches a particular severity.
Monitor Conditions (Check monitors only)	A monitor condition defines when a problem is raised.
Monitor Values (Check monitors only)	A monitor value is a numeric, string or timestamp value used to define monitor conditions.
Creation Time	The creation time of this monitor.
Last Modification Time	The last modification time of this monitor.

Showing Monitors as Graphs in the Dashboard

Use

To get a quick and intuitive overview of important or frequently used monitors, you can add them to the dashboard, which was designed to display monitors in form of graphs or tachometers. You must first have created a monitor tree as described in the Monitoring System Performance section. You can create dashboard dials for leaf nodes with integer values only, these have one or more values in the value column.

Dashboard dials can be moved up, down and sideways, allowing you to position them where you like using drag-and-drop. The way you set the dials has an influence on their size. You should always enter a minimum and a maximum value, which for CPU usage monitor, for example, would be `0` and `100` respectively. If you do not provide a maximum and a minimum, a scale will be generated based on the maximum and minimum value measured, this scale can be reset.

The following dial-types are available:

- Graph - a diagram showing current load and load-history
- Tacho - a speedometer-like dial that displays the current load
- History Tacho - a speedometer-like dial that displays the current load with graph dial built-in to show load-history
- Job Status Pie Chart - a circular diagram in 3d showing proportional representations

Actions

Action	Description
Reset Scale	Reset the computed scale
Edit	Edit the dial

Delete	Delete the dial
Dashboard > Edit	Edit the current dashboard
Dashboard > Delete	Delete the current dashboard with all dials on it
Create dashboard dial	Create a new dial from a monitor leaf-node

Procedure

To create a dashboard

1. Navigate to *Monitoring* > *Dashboard*.
2. Choose *Create* from the context menu and fill in at least the *Name* field.
3. Choose *Save & Close*.

Add a dial for a monitor leaf node

1. Navigate to *Monitoring* > *Dashboard* and select the desired dashboard from the drop-down list.
2. Navigate to *Monitoring* > *Monitors* and drill down the monitor tree until you have reached the desired leaf node.
3. Choose *Create dashboard dial* from the context menu of the monitor leaf node.
4. In the new Window, it is recommended to change the name with a descriptive *Label* as well as a desired dial *Type*, if you choose *Graph*, you should set the minimum and maximum values.
5. Choose *Save & Close* and navigate to *Monitoring* > *Dashboard* to verify the new dial gets displayed, you might have to refresh the tab. You can move the dial via its context menu.

To decrease the update interval of platform agent monitoring data

1. Navigate to *Environment* > *Process Servers*.
2. Choose *Stop* from the context menu of the process server belonging to the platform agent.
3. Choose *Edit* from the context menu of the same process server.
4. On the *Parameters* tab, add or modify the value of the process server parameter *MonitorInterval*, for example `3` seconds.
5. Choose *Save & Close*
6. Choose *Start* from the context menu of the process server.

Values

Field	**Description**	**Default Value**
Label	The display name of the dial	<The name of the monitor leaf node>
Type	The type of dial	Graph or tacho
Minimum	The minimum value of the scale	
Maximum	The maximum value of the scale	
History time	The time data is kept in the graph	

Monitor Value	The value that gets displayed	<The name of the monitor leaf node>

Monitoring SAP Systems

Use

Various monitors are available on each SAP System. SAP CPS has the ability to display monitors of different SAP Systems in one central tree. Monitoring of SAP Systems is done via the XAL interface, which allows access to remote monitoring data. The RFC user requires a specific set of privileges to use the XAL interface as outlined in the Reference SAP RFC User Privileges.

The following job definitions are available for interacting with SAP monitors:

- SAP_DeleteCcmsMonitors
- SAP_ImportCcmsMonitors

The RFC connection immediately retrieves the lists of available monitors from the SAP instance as soon as the XAL interface is configured for the SAP System.

For more information on monitors, refer to the Monitors section.

Interacting with Operator Messages

Use

Operator messages are generated when errors occur, for example when jobs reach the status `Error`. For jobs, this behavior can be customized in job chains with `status handlers` and with the *System_OperatorMessage* job definition. Operator messages are used to provide more information about errors that occur and/or ask the operator which steps to take when in a given situation.

The operator messages monitor gives an overview of all operator messages and their statuses. The operator messages monitor consists of two panes:

- The operator message monitor pane displays a list of all operator messages and their status that match a chosen filter. You can delete and reply to operator messages from the context menu.
- The operator message detail pane displays data of the currently selected operator message and the ability to delete and reply to operator messages.

The operator messages monitor is part of the Monitoring section in the navigation bar.

The possible reply states are:

- *NotRequired* - The operator message does not require intervention.
- *Required* - The operator message does require intervention and must be replied to.
- *Replied* - An operator has replied to the operator message.

Note

Empty replies are not accepted.

Reply expressions can be defined as regular expressions, which force the reply to match the regular expression.

Actions

Operator Messages support the following actions:

Action	Description
Reply	Reply to the Operator message (only available when the reply status is required)
Delete	Delete the Operator message
Expand All	Expand all Operator messages in the current filter
Filter > New Filter	Create a new Operator message filter
Filter > Edit Filter	Edit current Operator message filter
Filter > Delete	Delete current Operator message filter

Finding Operator Messages

You can search for operator messages using filters and the *Search Operator Messages* box on the Operator Messages tab. This box is known as the *intelliSearch* box and located under your username on the top right-hand side of the user interface. Filters allow you to specify a list of objects with static criteria. IntelliSearch allows you to specify complex queries in a simple way using prefixes. Prefixes are used to specify which property you are searching in and have short and long syntaxes. For example, if you want to display all operator messages created between 2:05 PM and 2:06 PM, you would use the search criteria as follows:

```
ct:14:05-14:06
```

You can search more than one property, as follows:

```
ct:14:05-14:06 rb:jdoe
```

Note

No spaces should be entered before or after the colon (*:*).

See the Advanced Object Search for more information.

The following table illustrates the available prefixes for operator messages:

Prefixes	Description
r, reply	reply text
rt, replytime	reply time
rb, repliedby	replied by user
ct, creationtime	creation time
lma, lastmodifiedat	last modification time
lmb, lastmodifiedby	last modified by user

The time-based prefixes allow you to use the time-specific syntax, see Advanced Object Search section for more information.

Procedure

To monitor an operator message:

1. Choose *Monitoring* > *Operator Messages*.
2. Choose the *Refresh* button to update the monitor.
3. Choose an operator message that has a `Reply Status` of `Required`.
4. Notice that information regarding this operator message is displayed in the detail pane.
5. From the context menu, choose *Reply* .
6. Depending on the operator message, either a word or phrase must be entered or a value must be chosen.
7. Choose or enter a reply.
8. Choose the *Save and Close* button to reply to the message and close the window.
9. Notice that the `Reply Status` has changed to *Replied*.

Values

- *Full Text* - The text of the operator message.
- *Reply status* - The reply status of the operator message, if it has been replied to or not.
- *Reply* - The reply the operator chose.
- *Sender Object* - The object that raised the operator message.
- *Creation Time* - The time the operator message was generated
- *Last Modification Time* - The last modification time

Example

Example 1

In the first example, an operator message is generated by a job chain named `EMEA_SALES_RUN` after it reached status `Error`, as defined in its Default job status handler. The operator navigates to the operator messages monitor and sees an operator message with the following information:

- *Creation Time* - 3:17:34 PM
- *Title Text* - Choose restart option
- *Reply Status* - Required
- *Reply* - [none, yet]

The operator selects the message and the operator message detail pane displays the following data:

- *Full Text* - Choose restart option
- *Reply Status* - Required

- *Sender Object* - EMEA_SALES_RUN_STEP_3
- *Creation Time* - 3:17:34 PM
- *Last Modification Time* - 3:17:34 PM

The operator now knows that a job chain job in step 3 reached status `Error` at 3:17:34 PM today, the step contains the name of the chain so the operator knows it is the `EMEA_SALES_RUN` job chain that failed. The operator would like to know what happened and chooses the Sender Object `EMEA_SALES_RUN_STEP_3`. After investigating the log file, the operator finds that the SAP System is down for maintenance. The operator returns to the operator messages monitor once the SAP System is live again and chooses *Reply* from the context menu of the operator message. The following replies are available:

- Abort chain and set chain status to "Error".
- Abort chain and set chain status to "Completed".
- Restart this step.
- Restart and Hold this step.
- Restart chain from the beginning.
- Rerun chain status handlers as if the step completed with status "Completed".
- Rerun chain status handlers as if the step completed with status "Error".

The operator chooses `Restart this step` and sees the following in the operator Message detail pane:

- *Full Text* - Choose restart option
- *Reply Status* - Replied
- *Sender Object* - EMEA_SALES_RUN_STEP_3
- *Creation Time* - 3:17:34 PM
- *Last Modification Time* - 3:27:55 PM

Example 2

In this second, more advanced example, a job chain uses operator messages to ask the operator questions. This is done via the job definition `System_OperatorMessage`.

An operator message with the following text is found in the operator messages monitor:

- *Full Text* - 'Has the error been corrected?'
- *Reply Status* - Required

The operator selects the operator message and chooses Reply from the context menu, a new window with the following information is displayed:

- *Full Text* - 'Has the error been corrected?'
- *Reply Expression* - `^The error has been corrected|The error has not been corrected$`
- *Reply* -

Note

to get a combo box in the reply window this syntax `^a|b|c|d$` must be used, it is a regular expression.

The '^' means `starting with`, the '|' means `or` and the '$' means `ending with`.

The Reply Expression displays the accepted replies in the form of a regular expression, in this example the operator has the choice between the following two replies from a combo box:

- 'The error has been corrected.'
- 'The error has not been corrected.'

The operator is certain the error has been corrected and chooses the following reply: `'The error has been corrected.'`

The operator now sees the following in the operator message detail pane:

- *Full Text* 'Has the error been corrected?'
- *Reply Status* Replied
- *Reply* 'The error has been corrected.'
- *Sender Object* Raise an operator message
- *Creation Time* 3:14:13 PM
- *Last Modification Time* 3:17:53 PM

Regular expressions are very powerful and can get complex. Avoid using special characters in the replyExpression unless their meaning in regular expression syntax is clear. Special characters include:

```
 '.', '?', '*', '+', '\', '^', '$', '{', '}', '|', '[[', ']', '(', ')'
```

Special characters can be escaped, which removes their special meaning, with the backslash `\` character. For example, you want to specify a path on Windows in a replyExpression, use double backlashes `\\` as path-separators, like `c:\\Windows\\temp`. Note that in this particular case you could also use forward slashes `/` instead, like `c:/Windows/temp`.

Receiving Operator Messages on your Desktop

Use

To receive and reply to operator messages on your Windows workstation without the use of a browser, you can install the operator messages monitor application, which will poll the Central Scheduler for operator messages. As soon as an operator message has been received which matches your criteria, it will be displayed in a pop-up window coming from the *system tray*, by default, the lower right-hand side of the *task bar*.

The application will remain in the *system tray* until you open it by choosing its *envelope* icon. The icon will have a red square when an operator message has been received which requires a reply.

There are three icons in the system tray disabled, warning and normal.

The application can be found in the *Configuration > Software* section and is named *opmsgmon-installer*, the installation only requires the choice of an installation directory. The installed software can be found in the *Scheduler* section of the Programs folder in the Windows Start Menu, the shortcut is named *Scheduler Operator Messages*.

The first time you launch the software, the configuration window is displayed, where you can fill in the URL as well as other connection details of the central server to subscribe to the feed.

The feed can have one of the following security settings:

- Viewable to all
- require a single password
- require a valid user name and password for the application server

The security settings can be configured in the SAP CPS registry using the registry key:

```
/configuration/feed/feeds/operator_message/password
```

- Not present - if the key is not present, there is no password and anyone can access the feed without a password
- <disabled> - the feed is disabled
- <login-only> - the feed is only accessible with a valid username and password
- any other value - any other value represents the password.

Note

The <> are part of the value, and not place holders. The following passwords are not allowed: `<disabled>` and `<login-only>`.

Note

The SAP CPS registry can be found in *Configuration > Registry*.

When an error occurs, there is a warning icon displayed on the main window of the application. When you choose the icon, you will get a new window with the exact error stack that was received from the application server. SAP support will request you to provide this information when you encounter issues.

Installer Arguments

The installer accepts the following argument:

- `-g filename` - where filename is the desired log file

This argument is used to troubleshoot installation issues you could encounter with the installer.

Prerequisites

- The full URL to the Central Scheduler system
- Depending on the security settings
  - the rss feed password

- login credentials for the Central Scheduler system

Procedure

Configuring the operator messages monitor

1. In the operator message monitor, navigate to *Tools* > *Configuration*
2. On the *Server* tab, select a protocol, fill in the *Hostname* and *Port* fields for the Central Scheduler.
3. Choose the 'Authentication" tab and fill in the applicable settings for security

Values

Tab	Field	Description	Default value
Server	Protocol	The protocol used by the application server, can be HTTP or the secure variant HTTPS	HTTP
Server	Hostname	The FQDN of the Central Scheduler, for example prd.masalan.com	
Server	Port	The Port of the Central Scheduler	50200
Server	Poll Interval	the interval at which requests for new operator messages should be made (in seconds)	30
Authentication	No password required	The Central Scheduler settings do not require a password to view operator messages	Y
Authentication	Use server username and password	The Central Scheduler settings require an account to access the operator messages	
Authentication	Use an RSS feed password	the Central Scheduler requires a specific password for operator message access	
Authentication	Isolation group	The isolation group of the operator messages that are to be displayed, for example *GLOBAL*.	
Messages	Show new messages for 5 seconds	The time the pop-up window is displayed	
Messages	List of rules to filter operator messages, only messages that match any filter are displayed		

Creating Advanced Reply Expressions

Use

Reply expressions of operator messages can be written as regular expressions [http://java.sun.com/javase/6/docs/-api/java/util/regex/package-summary.html], which have a syntax similar to that used by Perl. Regular expressions are used to match a character sequence in a string, here they are used to force valid answers and can be used to create lists of pre-defined answers.

With the use of regular expression, you can force operators to provide a reply that matches a specific pattern, the table below lists some of the characters that have a special meaning in regular expressions.

Note

Empty replies are not accepted!

Note

Spaces before and after the reply are trimmed. Multiple spaces and/or tabs in a sequence in the reply are replaced with one space. This means that replies consisting of spaces and/or tabs only are not allowed. It is not possible to answer a reply expression like `\s` or `\s+`.

Special character	meaning
.	Any one character (equivalent to '?' on Microsoft Windows systems)
*	Used in conjunction with another character, the previous character sequence repeated zero or more times.
+	Used in conjunction with another character, the previous character sequence repeated one or more times.
?	Used in conjunction with another character, the previous character sequence repeated once or not at all.
^	Starts with
|	The "or" character
$	Ends with
\	The escape character, used as prefix to specify any special character

The following table lists some special patters patterns:

Patterns	Description
\n	A newline character
\d	A digit
\D	Anything but a digit

The following constructs are frequently used:

- `^a|b|c$` - a combo-box which offers valid answers, "a", "b" or "c"
- `.*` - Any reply accepted (except spaces, null replies).
- `.+` - Any reply accepted (except spaces; the regular expression disallows null replies).
- `\d+` - Any number of digits, used to require a job id.

Integrating SAP CPS into your SNMP Landscape

Use

Historically, SNMP was used to monitor network devices; this has evolved as monitoring applications and processes becomes more and more important. Using the existing infrastructure and internal knowledge allows many companies to save costs related to the implementation of a application-specific monitoring infrastructure. SAP CPS allows you to send SNMP traps to your network management stations. If your management software supports loading `MIB` files we suggest that you install these in the SNMP receiving software. How this is done is dependent on the management software.

Note

SNMP integration requires the SNMP module, please check your license to make sure you have the *Module.SNMP* license key.

Network monitoring solutions retrieve the settings from MIB files to be able to communicate with devices and applications. The MIB file is a text file which contains information on the device or application and how to communicate with it. SAP CPS offers a MIB file which contains the definition of a trap. The standard MIB file complies with the standards and should easily be imported into network management system. You can edit this file, however, SAP strongly recommends you pay great attention when editing this file, as the MIB file must comply with syntax requirements.

The job definition *System_SNMP_SendTrap* can be submitted within any RedwoodScript job or job chain to send a trap to the SNMP monitor. Also, by using events or job chains, you can send such logical events within SAP CPS to the management software (for example, if an important job fails). It has two parameters, `TRAPID` and `TRAPDATA`, they can contain virtually any number or string that suit your corporate guidelines for your network management solution. Usually, customers use a fixed trap id and a variable message, containing information about which job failed. Note that you can use Redwood Expression Language to generate dynamic parameter values.

There are many different network management solutions, in the following example, we use HP OpenView; see the manual for your management software.

Example

Configuring HP OpenView

1. In SAP CPS, navigate to *Configuration* > *Software*.
2. Download the *redwood-mib.mib* file.
3. In HP OpenView, navigate to *[Options]* > *[Load/Unload MIB: SNMP]* > *[Load]* and choose the *redwood-mib.mib* file. When you are asked to load the definitions in the event system, choose *OK*. Now the Management Information Base is known to HP OpenView NNM, and traps from the process server can be received.
4. Navigate to *[Options]* > *[Event configuration]*. There you will see "RedwoodTraps" in the Enterprises view. When you choose that line, you will see more specific information in the events view. Only one event is specified: *userTrap*. This event is raised when you use the *System_SNMP_SendTrap* in SAP CPS. Choose the line to view the details of the UserTrap event.
5. In *[Event Message]* choose what you want to do with the trap. For example, choose the monitor where you want to log all the process server traps (for example *Application Alert Events*). Also, as an event log message you can give the following 2 specific messages:
    a. $1 gives the `TRAPID` parameter of *System_SNMP_SendTrap* (for example the job id).
    b. $2 gives the `TRAPDATA` parameter, the message that you specified.
6. On the *[Actions]* tab you can specify the message ($2) to be displayed in a pop-up window that is shown as soon as the trap is received. An example is: `net send <host> $2`

See Also

- Sending SNMP Traps

Sending an SNMP Trap

Use

You send SNMP traps to network management solutions to be able to report the status of specific jobs. SNMP is frequently used to monitor network devices and is more and more used to monitor applications, as the monitoring infrastructure is already present, no new solution has to be implemented.

Procedure

Sending a trap with a job chain

1. Navigate to *Definitions* > *Job Chains*.
2. Choose *Edit* from the context-menu of an existing job chain.
3. From the context menu of the top-level job chain, choose *Add Step*, give the step a user-friendly name to be able to easily locate it later.
4. From the context menu of the step, choose *New Job* and fill `SNMP` into the *Job Definition* field, select *System_SNMP_SendTrap*, specify default values for the parameters.
5. On every step, select *Goto Step* in the *Action* field and select the *notification* step for *On Canceled*, *On Error*, and *On Killed*.
6. Additionally, on the before last step, select *Mark Completed* in the *On Completed* field.
7. Choose *Save & Close*.

Example

Creating a job chain to send a trap

1. Navigate to *Definitions* > *Job Chains*.
2. Choose *Create* from the context-menu, fill `JC_SendTrap` into the *Name* field on the *Job Definition* tab.
3. On the *Job Chain* tab, choose *Add Step* from the context menu of the job chain and give the step the name *Retrieve System Info*.
4. Choose *Add Job* from the context menu of the step and fill `System_Info` into the *Job Definition* field.
5. From the context menu of the job chain, choose *Add Step*, give the step the name *Send Trap On Error*.
6. From the context menu of the step, choose *New Job* and fill `System_SNMP_SendTrap` into the *Job Definition* field.
   a. Fill `snmp.masalan.com` into *Host*, `162` into *Port*, `PUBLIC` into *Community*, *=jobId* into *ID* and `='The job '+jobId+' failed, upper-most job chain id is:'+topLevelJobId` into the *Data* field.
7. On the step *Retrieve System Info*, select *Goto Step* in the *Action* field and select the *Send Trap On Error* step for *On Canceled*, *On Error*, and *On Killed*.
8. Additionally, select *Mark Completed* in the *On Completed* field of the *Retrieve System Info* step.

9. Choose *Save & Close*.

Sending a test trap

1. Choose *Submit* from the context menu of the job chain *JC_SendTrap* (created in previous example).
2. On the *Times and Dates* tab, set the *Requested Start Time* 5 minutes in the future and choose *Submit*.
3. Navigate to *Monitoring* > *Jobs* and locate the job chain you submitted previously, expand it fully and cancel the *Generate an overview of the system* job of the chain.
4. Wait for the job chain to finish and open your network management solution to inspect the trap.

See Also

- Integrating SAP CPS into your SNMP Landscape

Integrating SAP CPS into Third-Party Applications

Use

The permalink API allows third-party applications to interact with SAP CPS and create, display, edit, and submit SAP CPS objects, as long as the object allows the action.

You need to have the PermaLink module, which requires the *Module.Scripting* license key.

Syntax

```
api-permalink/(list|show|edit|create|submit|help)?link-objecttype=<objecttype>&(link-path=<name>|link-uniquei
```

- <objecttype> - The object type; not all object types are supported. For example, `JobDefinition`
- <name> - The full name of the object in the form of <partition>.<objectName>, for example `GLOBAL.System_Info`
- <n> - The unique Id of the object.

Prerequisites

- *Module.Scripting* license key

list

Show the overview page for a certain object type

Example

Open JobDefinition overview

```
http://pr1.masalan.com:50200/scheduler/api-permalink/list?link-objecttype=JobDefinition
```

show

Open a show page for the given object

Job Files

For linking to the JobFile show pages, there are extra convenience attributes, which allow to access the JobFile show page by: job id and (filename or file order).

Example

Open JobDefinition System_Info show page by business key

```
  http://pr1.masalan.com:50200/scheduler/api-permalink/show?link-objecttype=JobDefinition&link-path=GLOBAL.Sy
```

Open JobDefinition System_Info show page by unique id

```
    http://pr1.masalan.com:50200/scheduler/api-permalink/show?link-objecttype=JobDefinition&link-uniqueid=790
```

Show JobFile page by unique id

```
  http://pr1.masalan.com:50200/scheduler/api-permalink/show?link-objecttype=JobFile&link-uniqueid=340644
```

Show JobFile page by job id and filename

```
http://pr1.masalan.com:50200/scheduler/api-permalink/show?link-objecttype=JobFile&link-jobid=22727&link-jobfi
```

Show JobFile page by job id and file order

```
http://pr1.masalan.com:50200/scheduler/api-permalink/show?link-objecttype=JobFile&link-jobid=22727&link-jobfi
```

Note

The attributes: link-jobid, link-jobfileorder and link-jobfilename *only* work for jobfiles!

edit

Open the edit window for the given object, besides in the main window the show

page is also shown for the object.

Example

Open JobDefinition System_Info edit page by business key

```
http://pr1.masalan.com:50200/scheduler/api-permalink/edit?link-objecttype=JobDefinition&link-path=GLOBAL.Syst
```

Open JobDefinition System_Info edit page by unique id

```
http://pr1.masalan.com:50200/scheduler/api-permalink/edit?link-objecttype=JobDefinition&link-uniqueid=79036
```

create

Open a create new object window for the given object type.

Example

Create a new format window.

```
http://pr1.masalan.com:50200/scheduler/api-permalink/create?link-objecttype=Format
```

submit

Open the submit window, optionally with a predefined JobDefinition.

Example

Submit a new Job and select the JobDefinition manually

```
  http://pr1.masalan.com:50200/scheduler/api-permalink/submit
```

Submit a new Job for the given JobDefinition

```
  http://pr1.masalan.com:50200/scheduler/api-permalink/submit?link-path=GLOBAL.System_Info
```

help

Since business keys are preferred for usage in permalinks, when you do not know the valid business key path for an object. You can use the help function to obtain the correct path for an object for which you do know the unique id.

Example

```
http://pr1.masalan.com:50200/scheduler/api-permalink/help?link-objecttype=JobDefinition&link-uniqueid=79036
```

This will present a show permalink to the JobDefinition using the business key, but only when the object type has a business key.

```
http://pr1.masalan.com:50200/scheduler/api-permalink/show?link-objecttype=JobDefinition&link-path=GLOBAL.Syst
```

Chapter 10. Scheduling Processes Outside of the SAP System Landscape

Use

The Chargeable version of SAP CPS has many more features than the Free of Charge version, opening a platform independent world of new possibilities.

SAP CPS not only allows you to schedule your SAP processes, but also all you Unix and Windows processes as well as other processes on other platform.

Implement SAP CPS and have a *Central Point of Control* for your entire data center with all your processes in one view.

Migrate your cron jobs or other scheduled tasks on the OS by defining job definitions, automate complex tasks with RedwoodScript jobs or trigger processes on your other external systems.

Note

Please check your license to see which platforms and external applications you can manage from within SAP CPS.

1. Using Platform Job Definition Types
2. IBM z/OS Job Definition Types
3. IBM AS/400 Job Definition Types
4. Creating Platform Process Servers
    a. Command Line System Tools
5. Creating AS/400 Process Servers

See Also

- Creating and Editing Job Definitions
- Creating and Editing Job Chains

Using Platform Job Definition Types

Use

Platform job definition types define a type of job definition and the service required to execute the jobs of that type on external systems. When job definition types are assigned to a process server, they define the type of jobs that the process server can run and services that are needed. You must only submit job definitions of a certain job definition type into queues that are handled by process servers capable of running those job definition types, or those jobs can never be started.

Note

Platform job definitions are part of the Platform module and require a specific license, please check the License section for more information.

Job definition types only support the basic action *Refresh*. It is not possible to *Create*, *Edit* or *Delete* job definition types.

The server has knowledge of which agents and process server services are capable of executing which job definition types in theory. Some combinations are allowed that do require optional software on the actual operating system that it will run on. For instance, the platform agent for an operating system is capable of starting Perl, but this does require that the Perl interpreter is installed on that particular system. The same holds for the UNIX job definition types such as BASH, CSH and KSH. They will only run successful if the appropriate shell is actually installed. These UNIX job definition types can also run on Microsoft Windows servers with one of the many 'UNIX emulation' packages available.

Note

Please pay attention when you mix Redwood job definition types with external ones, see Process Server Services for more information.

Note

When you create or edit a process server, the required services are added automatically to the process server based on the job definition types you configured, so there is no need to change services on a process server. The only exception is for process servers that are used for file events exclusively, there you do not need to specify a job definition type, however, the *PlatformAgentService* has to be added manually.

To view a job definition type, choose a job definition type in the overview table. The job definition type details are shown in the lower detail pane.

The following standard job definition types are provided with SAP CPS:

UNIX job definition types

- BASH - For writing Bourne Again SHell scripts. BASH is installed on all Linux systems and optionally on all UNIX (compatible) platforms that SAP CPS runs on.
- CSH - For writing C Shell scripts. Available on most UNIX systems.
- KSH - For writing Korn Shell, for job definitions that run on UNIX (compatible) servers.
- PERL - For writing Perl scripts.
- PERL (Unicode) - For writing Perl scripts with UNICODE characters in your script source.

Microsoft Windows job definition types

- CMD - For writing Microsoft Windows Command scripts.
- PS1 - For writing Microsoft Windows PowerShell language scripts. PowerShell is available as a download for Microsoft Server 2003, XP and Vista and is delivered with Microsoft Server 2008.
- VBScript - For writing Microsoft Windows Visual Basic Script job definitions.

Note

All UNIX job definition types will also run on Windows, provided you have the command interpreter installed.

See Also

- Different Versions
- Checking Your License

Unix Job Definition Types

Using the BASH Job Definition Type

Use

The BASH shell is the default UNIX shell on Linux, and is usually available on other UNIX systems such as AIX, HP-UX and Solaris. Various ports also exist for Microsoft Windows.

Note

You must assign at least one process server to run BASH job definitions in order to use the job definition type.

Variables and Parameters

- Parameters in the job definition are manipulated in the BASH source simply as variables, using the standard `$PARAMETER` syntax.
- The BASH shell does not have actual data types; all parameter values are stored as strings. Numbers are translated automatically. Dates are sent and retrieved using the Script Date Format.
- Out parameters are supported by setting the parameter using the PARAMETER=VALUE syntax.

Returning an Error

If your script code exits with a non-zero exit code this is correctly reflected in the server.

Code a `set -e` command to abort script execution immediately on running a command that has a non-zero exit status.

BASH Examples

The following shows how to use a normal environment variable:

```
echo This is running under user $USER with home directory $HOME and temporary directory $TMPDIR.
```

The following shows how to pass numeric (N1), string (S1) and date (D1) parameters to and from KSH scripts:

```
# Note BASH allows no spaces around = in variable assignment
N1=$((expr $N1 + $N1))  # Add numeric variable to itself
S1="$S1 $S1"  # Concatenate string
DTEMP="1999/12/31 23:59:59,000 GMT"
echo You said $D1, I propose $DTEMP
```

```
D1=$DTEMP   # Set DateTime to new string value
```

The following example shows you return a non-zero exit code resulting in the job going into Error:

```
N=1
echo "Exiting with value $N."
exit $N

echo "Not reached"
```

Using the CSH Job Definition Type

Use

The C shell is an alternative UNIX shell on various UNIX systems such as AIX, HP-UX and Solaris. It is also available on Linux and Microsoft Windows.

For compatibility with the KSH job definition type on Windows and Linux, you must install `tcsh` separately if it is not already installed.

Note

You must assign at least one process server to run CSH job definitions in order to use the job definition type.

Should you use C Shell scripts

CSH is not recommended to code new scripts in. It is not possible to store all possible values in C Shell variables, and it does not support functions or other structured programming concepts very well. A well known article explaining this in more detail can be found by searching the internet for 'CSH programming considered harmful'.

We still offer the CSH job definition type for those cases where you have historic code written in the C shell that needs to be executed as a job. For new code, use the BASH, KSH or Perl job definition types.

Variables and Parameters

- Parameters in the job definition are manipulated in the CSH source simply as variables, using the standard `$PARAMETER` syntax.
- The CSH shell variables do not have actual data types; all parameter values are stored as strings. Numbers are translated automatically. Dates are sent and retrieved using the Script Date Format.
- Out parameters are supported by setting the parameter using the `set PARAMETER=VALUE` or `@ PARAMETER = VALUE` syntax.

Returning an Error

If your script code exits with a non-zero exit code this is correctly reflected in the server. When a CSH script returns with a non-zero error code, the output parameters are *not* set.

CSH Examples

The following shows how to use a normal environment variable:

```
echo This is running under user $USER with home directory $HOME and temporary directory $TMPDIR.
```

The following shows how to pass numeric (N1), string (S1) and date (D1) parameters to and from KSH scripts:

```
# Add numeric variable to itself
@ N1 = $N1 + $N1
# Concatenate string
set S1="$S1 $S1"
set DTEMP="1999/12/31 23:59:59,000 GMT"
echo You said $D1, I propose $DTEMP
# Set DateTime to new string value
set D1="$DTEMP"
```

The following example shows you how to return a non-zero exit code resulting in the job going into Error:

```
set N=1
echo "Exiting with value $N."
if ( $N > 0 ) then
  exit $N
endif
echo "Not reached"
```

Using the KSH Job Definition Type

Use

The Korn shell is the default UNIX shell on various UNIX systems such as AIX, HP-UX and Solaris. It is also available for Linux and Microsoft Windows in the form of `pdksh`.

Note

You must assign at least one process server to run KSH job definitions in order to use the job definition type.

Writing portable KSH scripts

KSH is not the same implementation on all platforms. We have found the following compatibility issues between these variations:

- Some versions of KSH do not support aliases in non-interactive invocations of the Korn shell. In other implementations, aliases can only be used in scripts that are not sourced, but executed. Since the job definition script code is sourced, it cannot call any aliases. We highly recommend that you use functions instead.
- On Solaris, the syntax `$(command)` to execute a program and substitute its standard output is not allowed. You should use a pair of grave accents to remain portable to Solaris: `` `command` ``.
- The `pdksh` does not implement all the features of AT&T ksh, and there are minor differences between the two. Please refer to the notes page of the pdksh website (http://www.cs.mun.ca/~michael/pdksh/NOTES) for a list of differences.

Variables and Parameters

- Parameters in the job definition are manipulated in the KSH source simply as variables, using the standard `$VARIABLE` syntax.
- The Korn shell does not have actual data types; all parameter values are stored as strings. Numbers are translated automatically. Dates are sent and retrieved using the Script Date Format.

- Out parameters are supported by setting the parameter using the `VARIABLE=VALUE` syntax.

Returning an Error

If your script code exits with a non-zero exit code this is correctly reflected in the server.

Code a `set -e` command to abort script execution immediately on running a command that has a non-zero exit status.

KSH Examples

The following shows how to use a normal environment variable:

```
echo This is running under user $USER with home directory $HOME and temporary directory $TMPDIR.
```

The following shows how to pass numeric (N1), string (S1) and date (D1) parameters to and from KSH scripts:

```
# Note KSH allows no spaces around = in variable assignment
N1=`expr $N1 + $N1`  # Add numeric variable to itself
S1="$S1 $S1"  # Concatenate string
echo "Concatenated string" $S1
DTEMP="1999/12/31 23:59:59,000 GMT"
echo You said $D1, I propose $DTEMP
D1=$DTEMP   # Set DateTime to new string value
```

The following example shows you how to return a non-zero exit code resulting in the job going into Error:

```
N=1
echo "Exiting with value $N."
exit $N

echo "Not reached"
```

Using the Perl Job Definition Type

Use

Perl is a widely used, multi-platform scripting language. SAP CPS includes two Perl job definition types, one of which uses unicode UTF-8 encoding for the source.

Note

You must assign at least one process server to run PERL job definitions in order to use the job definition type.

A Perl interpreter needs to be installed on the system that the platform agent runs on. On UNIX systems it should be installed as `/usr/bin/perl`. On Microsoft Windows systems the Perl interpreter will be found via the `%PATH%` environment variable. For UNICODE to work properly the version should be 5.8.2 or higher.

Variables and Parameters

- Parameters to the script are manipulated in the PERL source simply as if they are variables, using the standard `$PARAMETER` syntax.

- Perl is a weak type-casted language; it only supports a single scalar data type: scalar. This maps directly for strings and numbers, but dates are subject to format conversions; see Scripting Date Formats.
- Out parameters are supported. Set the variable using any Perl method. For example a simple assignment in Perl is `$PARAMETER = VALUE;`.

Returning an Error

If your script code exits with a non-zero exit code this is correctly reflected in the server. Note that the maximum value that can be returned is operating system dependent; we recommend not relying on more than one byte (0..255).

Perl Examples

The following shows how to use a built-in and an environment variable as well as job definition parameter:

```
print "Hello from " . $^O . "!\n";            # built-in variable
print "OS user = " . $ENV{USER} . "\n";       # environment variable
print "Your message was " . $message . "\n";  # job definition parameter
```

Match one variable to a pattern, the following job definition needs two string parameters defined: str and pattern.

```
if ($str =~ m/$pattern/)
{
  print "Match $i& found!\n";
}
else
{
  die "$str does not match $pattern...\n";
}
```

The following example shows you return a non-zero exit code resulting in the job going into Error:

```
{
  $N = 1;
  print "Exiting with exit code $N.\n";

  return $N;

  print "Not reached.\n";
}
```

Using the SQLPLUS Job Definition Type

Use

SQLPLUS is a command line SQL and PL/SQL language interface and reporting tool that ships with Oracle.

The process server needs to have the *SQLPLUS* job definition type, this is not set by default on platform agent process servers.

The platform agent host needs to have Oracle Software installed with an `sqlplus` binary and the Oracle environment must be set.

Variables and Parameters

- Parameters to the script are used in the SQLPlus source as defines, using the standard `&PARAMETER.` syntax.

- Since SQLPlus defines are always string based, all parameters are thought of as strings. DCL strings are limited to 255 characters.
- OUT parameters are supported; just define a variable with the name of the parameter using the regular method. To set an output parameter use `define PARAMETER = 'VALUE'`. If you are using date parameters, they must be in the standard Oracle date format, `SYYYMMDDHH24MISS`.

Returning an Exception

If your PL/SQL code exits with an exception, this is ignored in the server unless you put `whenever sqlerror exit failure` into your code.

SQLPLUS Examples

The following example shows the built-in parameters being used as symbols and how to set an output parameter. Note that the job definition needs an *Out* parameter named `P_BYTESFREE`.

```
rem
rem Shows built-in variables and
rem computes output parameter.
rem

column bytes new_value P_BYTESFREE

select sum(bytes) "BYTES"
from   dba_free_space;

prompt Returning free space value: &P_BYTESFREE
```

The following example shows how to hide the "Connected." output that is normally generated by SQL*Plus if you connect to a database whilst spooling.

```
spool output.txt
prompt Logged in as &_USER...

spool off
connect scott/tiger@masalan.com

rem "append" requires 10g or later

spool output.txt append

prompt Look, no output above!

prompt Logged in as &_USER...
```

Microsoft Windows Job Definition Types

Using the CMD Job Definition Type

Use

The Microsoft Windows Command CMD language is the Windows version of BAT (MS-DOS batch) files. It supports everything that BAT supported, and more. The CMD language is quite powerful nowadays; of particular interest when writing scripts are the SET, IF and FOR commands. Type `HELP` followed by a command in a command prompt for more information.

Note

You must assign at least one process server to run CMD job definitions in order to use the job definition type.

For commands that interact with COM or WMI commands it is better to use the VBScript or PowerShell job definition types.

Variables and Parameters

- Parameters to the job definition are manipulated in the CMD source simply as if they are variables, using the standard %PARAMETER% syntax.
- The CMD language does not have actual data types; all parameter values are stored as strings. Numbers are translated automatically. Dates are sent and retrieved using the Script Date Format.
- Out parameters are supported by setting the parameter using the `set PARAMETER=VALUE` syntax.

Returning an Error

If the CMD file ends by means of the EXIT command, the output variables will not be set. The correct ways to end the job definition and still set the output variables are:

- Running to the end of the job definition.
- Use GOTO :EOF (see HELP GOTO).
- Use EXIT /B.

Examples

The following shows how to pass numeric (N1), string (S1) and date (D1) parameters to and from CMD scripts:

```
set /A N1=%N1% + %N1%
rem Concatenate string
set S1=%S1% %S1%
set DTEMP=1999/12/31 23:59:59,000 GMT"
echo You said %D1%, I propose %DTEMP%
rem Set DateTime to new string value
set D1=%DTEMP%
```

The following example shows you return a non-zero exit code resulting in the job going into Error:

```
N=1
echo Exiting with value %N%.
if %N% GEQ 1 exit/b %N%

echo Not reached
```

The following shows how to obtain the number of free bytes on drive C on the process server that the job runs on:

```
rem
rem Find out how many bytes are free on drive C:, and set the DIRSPACE variable
rem accordingly. This depends on "dir" writing the drivespace on its last line
rem of output. We continuously overwrite the same variable until we automatically
rem leave the last value in place.
rem
```

```
for /F "usebackq tokens=3" %%x in (`dir /-C c:\`) do call :getline %%x
goto :EOF

:getline
set DIRSPACE=%1
goto :EOF
```

Using the PowerShell Job Definition Type

Use

The Microsoft Windows PowerShell is a new shell meant for interactive use that can also be used for non-interactive scripts. It comes as standard with Microsoft Server 2008, and can be downloaded for Microsoft Windows XP, Vista and Server 2003.

Note

You must assign at least one process server to run PowerShell job definitions in order to use the job definition type.

The PowerShell is very strong in interacting with .Net objects, as it can pass objects between different 'cmdlets' instead of strings, as in most shells.

To make PS1 scripts available for script writing, install the PowerShell executable so that it is available on the path.

Note

PowerShell also supports regular expressions, however, the implementation is very different in Powershell when you compare it to other tools. Please make sure you test your regular expressions extensively!

Execution Policy

The default execution policy of the PowerShell is *Restricted*, which does not allow the execution of scripts.

SAP CPS supports the following two policies:

- *RemoteSigned*
- *Unrestricted*

Run `get-help about_signing` in the PowerShell to learn how to set the desired execution policy.

Variables and Parameters

- Parameters to the job definition are manipulated in the PS1 source simply as if they are variables, using the standard $VARIABLE syntax.
- The PowerShell language has knowledge of data types. The String and Number data types are supported. DateTimeZone parameters are stored in DateTime variables. This means that you cannot transfer the time zone of a parameter to a PowerShell variable; it will be converted to the corresponding date-time in the local time zone of the Microsoft Windows system. Out parameters are converted to UTC (GMT).
- Out parameters are supported by setting the parameter using the `$VARIABLE = VALUE` syntax or the `Set-Variable`, `New-Variable` cmdlets.

Returning an Error

If your script code exits with a non-zero exit code this is correctly reflected in the server, and output parameters are set as far as they were already set.

Examples

The following shows how to pass numeric (N1), string (S1) and date (D1) parameters to and from PS1 scripts:

```
$N1=$N1 + $N1
# Concatenate string
$S1="$S1 $S1"
[datetime]$DTEMP="1999/12/31 23:59:59Z"
echo "You said $D1, I propose $DTEMP"
$D1=$DTEMP
```

The following example shows you how to return a non-zero exit code resulting in the job going into Error:

```
$N=1
echo "Exiting with value $N."
if ($N > 0)
{
  exit $N;
}

echo "Not reached"
```

The following example shows how to rename all the files in a directory:

```
ls | foreach {ren $_.fullname $_.name.replace("BI1", "BI2")}
```

Using the Visual Basic Script Job Definition Type

Use

The Microsoft Windows VBScript (VBS) language is a variation of Visual Basic, and is very suited for writing scripts. The `cscript` command-line interpreter is available on all recent Microsoft Windows server operating systems.

Note

You must assign at least one process server to run VBS job definitions in order to use the job definition type.

Variables and Parameters

- Parameters to the job definition are manipulated in the VBScript source simply as if they are variables, using the standard syntax.
- The VBScript language has knowledge of data-types. The String and Number data-types are supported. DateTimeZone parameters are stored in string variables.
- Out parameters are supported by setting the global variable with the same name as the parameter to the desired value.

Returning an Error

If your script code needs to set a non-zero exit code you usually code `Wscript.Quit(n)`. This is correctly reflected in the server, but it will not set any output variables. To set output variables **and** stop execution, call the global sub `Quit(n)` instead, as seen in the example below. This sub is only available in the context of SAP CPS jobs.

Your code starts out with error handling set to abort execution. When a runtime error occurs, the job will fail with *Error* and return-code 255. When you decide to handle errors using the `resume next` you must make sure that you clear the error after you have handled an error, or at least before you end the script. If you 'fall off the bottom' by not calling `Wscript.Quit` or `Quit` and the `Err` object has not been reset your script will end in the *Error* status.

Examples

The following example shows how to pass numeric (N1), string (S1) and date (D1) parameters to and from VBS jobs:

```
N1 = N1 + N1
S1 = S1 & S1
DTEMP = "1999/12/31 23:59:59Z"
Wscript.stdout.writeline "You said " & D1 & ", I propose " & DTEMP
D1 = DTEMP
```

The following example shows you return a non-zero exit code resulting in the job going into Error, as well as setting any output variables:

```
N = 1
Wscript.stdout.writeline "Exiting with value " & N
Quit N
```

HP OpenVMS Job Definition Type

Use

The HP OpenVMS job definition type allows you to schedule workload on HP OpenVMS. This functionality is not available when you purchase SAP CPS from SAP AG directly, please contact your account representative if you require this job definition type.

1. Using the DCL Job Definition Type

See Also

- Different Versions
- Checking Your License

Using the DCL Job Definition Type

Use

The DCL shell is the default, and usually the only, shell on HP OpenVMS. It is used for almost all scripting on VMS systems, as it is guaranteed to be installed and there is a low likelihood of any other shell being installed. Even Perl is not very popular on OpenVMS (although support for this is anticipated for a future release).

Variables and Parameters

- Parameters in the job definition are manipulated in the DCL source as *global symbols*, using the standard `SYMBOL` or `'SYMBOL` syntax.
- All parameters are passed as string values. Numbers are translated automatically. Dates are sent and retrieved using the Script Date Format.
- Out parameters are supported.

Note that your DCL script runs as a nested DCL script, with its own local symbols P1 through P8 that are not set to any value. This means that if you use job definitions that use the parameters P1 through P8 that you must delete the local symbol before you can access the global symbol value.

For instance, to print the value of the parameter P1 use:

```
$ delete/symbol/local P1
$ write sys$output "The parameter P1 = ''P1''
```

Returning an Error

The DCL $status is correctly reflected in the Repository, and the job status is also derived using the normal VMS method. Note that VMS exit statuses often contain large number values that can still indicate success.

Standard files

Your DCL code runs with it's default directory set to `[...<job>.JobFiles]`. Any files that it creates there will be

attached to the job. This directory normally contains three standard files:

- STDOUT.LOG;1 which will contain any output sent to SYS$OUTPUT, including any DCL verify output written by your script.
- STDERR.LOG;1 which will contain any output sent to SYS$ERROR, including any DCL verify output written by the DCL job definition runner and the `SYLOGIN.COM` and `LOGIN.COM` procedures, as well as the trailing job information written by the command processor when the job finishes.
- STDLOG.LOG;1 which will contain any debugging and informational output created by the `job-processor`. If you are not running the process server with logging at debug level this file will usually be empty and be deleted automatically.

In most VMS installations, `SYLOGIN.COM` contains this in the first two lines:

```
$ Set NoOn
$ VERIFY = F$VERIFY(F$TRNLNM("SYLOGIN_VERIFY"))
```

As the SAP CPS platform-agent starts logging from a very early stage the `STDERR.LOG` file may contain the above output. To remove this output, change the `SYLOGIN.COM` file to start as follows:

```
$! VERIFY = 'F$VERIFY(F$TRNLNM("SYLOGIN_VERIFY"))
$ Set NoOn
```

Note the added comment and quote characters, and the reversed order. The `$!` line is not *logged* but it is executed by DCL as it contains the magic incantation `'F$VERIFY`.

See Also

- Different Versions
- Checking Your License

The Environment of Platform Jobs

Use

All OS jobs that run on the platform agent obtain a reliable environment. The environment of the job is kept as consistent as possible, regardless of OS and/or OS family (UNIX, Microsoft Windows, HP OpenVMS)

Some platforms are differ slightly, especially where the native job environment is re-used (IBM z/OS, for example).

For UNIX and Windows, a job's environment contains the following:

- account - the account used to run the code
- files - output and log files
- environment variables - standard environment variables and SAP CPS-specific ones that store the job id, for example.

Note

You must install the SAP CPS platform agent on a local file system; SAN file systems might be considered local, when they are mounted as iSCSI, for example.

Account

The job will be run under the account that is specified in the job definition. If this is not null but the account is not available the job will fail (status Error). If the account is null, a default account will be used.

Open files

The current directory for the job is a directory where you can place files. All files placed there will be listed as job output files, and are viewable in the user interface.

- A job can write text output to `stdout` and `stderr`.
- Reading from `stdin` will result in no data being returned.
- No other file handles or descriptors are open.

Environment Variables

UNIX

For UNIX the set of environment variables is controlled by the `KeepEnvironmentVariables` process server parameter.

Environment variables are inherited from the environment that is defined for the `network-processor`.

Additionally the following variables are set:

- JCS_JOB_ID - set to the JobId of the job.
- JCS_JOB_FILES_DIR - points to the directory that the job can write files to that will be listed as job output files. The job also starts out with this directory as its current directory.
- SHELL, HOME, LOGNAME and USER - set to the correct value of the account that the job runs as.
- TMPDIR, TEMP and TMP - point to a directory that contains job-specific temporary files. The job can add extra temporary files there. They will be removed when the job finishes.
- PATH - inherited from the network-processor, but prefixed with the location of the SAP CPS tools. This way you can call tools such as jsleep, jftp, jevent, jshell without having to specify a path. If PATH is not in the KeepEnvironmentVariables parameter, it will be set to a default path consisting of the location of the SAP CPS tools, /usr/bin and /bin.

Microsoft Windows

For Microsoft Windows the environment variables are formed by recreating the standard set of environment variables for the user that the job runs for, by looking these up in the user hive in the registry. This is equivalent to the values that you would find active when you would log on interactively as that user.

Additionally the following environment variables are set:

- JCS_JOB_ID - set to the JobId of the job.
- JCS_JOB_FILES_DIR - points to the directory that the job can write files to that will be listed as job output files. The job also starts out with this directory as its current directory.
- JCS_CODEPAGE - set to the value of the ANSI code page in effect for the agent. All job definition type runner scripts select this code page as the OEM (DOS) code page, so that all local files are written in this code page.
- USERNAME, USERDOMAIN and USERPROFILE - set to the correct value of the account that the job runs as.
- TMPDIR, TEMP and TMP - point to a directory that contains job-specific temporary files. The job can add extra temporary files there. They will be removed when the job finishes.
- PATH - will be set from the user hive, but prefixed with the location of the SAP CPS tools. This way you can call SAP CPS tools such as jsleep, jftp, jevent, jshell without having to specify a path.

Using External Job Definition Types

Use

The following external job definition types are available:

1. IBM z/OS Job Definition Types
    a. Using the JCL_FTP Job Definition Type
    b. Using the JCL Job Definition Type
2. IBM AS/400 Job Definition Types
    a. Using the AS/400 Job Definition Type

See Also

- Different Versions
- Checking Your License

IBM z/OS Job Definition Types

Use

There are two job definition types for the mainframe, JCL_FTP and JCL. JCL_FTP uses the FTP interface on the mainframe to start jobs, this has the advantage of not requiring additional software on the mainframe. The JCL job definition requires a SAP CPS process server to be installed on the zSeries server. The zSeries connector is not available from SAP AG, if you require this functionality, please contact your account representative.

1. Using the JCL_FTP Job Definition Type
2. Using the JCL Job Definition Type

See Also

- Different Versions
- Checking Your License

Using the JCL_FTP Job Definition Type

Use

The JCL_FTP job definition type works with a *zero footprint agent*. In other words, no SAP software needs to be installed on the mainframe. The agent that performs the submit to JES via a (remote) internal reader runs on a UNIX or Microsoft Windows system. JES2 or JES3 processes the job itself. As with all job definition types you must assign at least one process server the JCL_FTP job definition type in order for that process server to perform the task and be assigned. The actual work of *executing* the job is done fully by the mainframe itself.

When SAP CPS submits the job it first sends it to (one of) the platform agent(s) that is assigned the JCL_FTP job definition type. On that server, the job processor runs and contacts the z/OS server over TCP/IP via the FTP protocol. It switches to the JES interface instead of the normal file interface and submits the job to JES. Once the job is submitted, it polls to see if the job is finished. Once the job is finished, it attempts to retrieve the jobs output datasets and puts each of them into a different SAP CPS job output file. It also translates the JES job outcome into a SAP CPS job status (ERROR or COMPLETED).

Note

You require the *ProcessServerService.JCL_FTP* license key in order to be able to use the JCL_FTP job definition type.

The source of a JCL_FTP job definition contains either a complete job description consisting of JCL, or a single line naming a DSN containing the JCL on the mainframe. A JCL coding starts with a line beginning with `//`. A DSN contains just the DSN name, without any keywords.

Example of JCL in the job definition source

```
//EXAMPLEX JOB (ACCT#), 'JOB DEMO',CLASS=X
```

```
//STEP1 EXEC PROG=IEFBR14
```

Example of a DSN PDS naming the JCL on the mainframe

```
HLQ.DEMO.CNTL(JCLDEMO)
```

Parameter substitution

One of the advantages that most job definition types have is that they use native parameter substitution. The source that you type as the job definition source can use all parameters that you defined on the job definition, plus some built-in ones, in the native syntax of the computer language that you are writing in.

The syntax that SAP CPS uses for parameters is:

~parameter~

For example, if you have parameters P_DSN1 varchar2(30), P_DSN1_TRACKS number(3), you can use this in the JES source like this:

//DSN1 DD DSN=~P_DSN1~,DISP=(NEW,CATLG),SPACE=(TRK,(~P_DSN1_TRACKS~,1))

Note

We use the tilde (~) character to start and end a parameter name, since this character was not represented in the earlier EBCDIC character sets, and thus not very likely to occur in normal JCL cards. However, current EBCDIC tables usually do have tilde, as code point A1, but it can only occur in '*' SYSIN cards, as JES does not allow the tilde character.

JOB card

Every job received by JES must start with a job card with the following format:

```
//jobname JOB (accounting),name,keyword-parameters
```

SAP CPS automatically generates a job card if it sees that you have not created one at the start of your code. This default card is:

```
//~JES_USER~~JES_JOB_IDENTIFIER~ JOB ~JES_ACCOUNT~,'~JES_USER~'[,CLASS=~JES_CLASS~][,MSGCLASS=~JES_MSGCLASS~]
```

The `,CLASS` and `,MSGCLASS` arguments are only added if they are filled by a job definition parameter.

SAP CPS communicates to z/OS using the IBM FTP server, which has a direct link to the JES internal reader and the JES spool. If the *JESINTERFACELEVEL* is 1 you must use a job name of `~JES_USER~` followed by one character. The default job card does this correctly. If the FTP server is running at `JESINTERFACELEVEL=2` then the job name is free, and the automatic job card will use a jobcard starting with `J` followed by a

Defining the user account

The job is submitted to MVS using an FTP account; this account will also be used to submit the job (unless overruled by the job card). The username, password and hostname where the job is submitted are normally set via *credentials*. Set the *RunAs* field on the job definition to define all three parameters.

Example

To submit a job to account MYUSER with password SECRET on host ZPROD1 use the following syntax in the RunAs field (or via a {VirtualUser}: reference to a stored credential set):

```
MYUSER/SECRET@ZPROD1
```

Note that as soon as you store this field the password is stored in an encrypted fashion.

Built-in Parameters

The JCL_FTP job definition type supports the following parameters:

Parameter Name	Description
JES_HOST	The z/OS hostname where the job is submitted (not recommended; use credentials instead).
JES_USER	The z/OS user account used to submit the job (not recommended; use credentials instead).
JES_PASSWORD	The z/OS user account password used to submit the job (not recommended; use credentials instead).
JES_JOB_IDENTIFIER	The job class. If This is not set, a random letter between A and Z will be taken (only used if no job card in JCL found).
JES_DELETE	Delete (cancel) the JES job after it ran?
JES_ACCOUNT	The accounting rule (only used if no job card in JCL found).
VERBOSE	How much logging to show on the agent side. The default = 0, but it can be set to 1 or 2 for debugging purposes.

Prerequisites

On the z/OS or MVS server:

- z/OS (any release) or OS/390 V1R2.0 or higher, preferably V2R10 or higher to support arbitrary job names.
- TCP/IP V3R1 or higher.
- IBM FTP server enabled.

On the UNIX or Microsoft Windows server that the platform agent runs on:

- Perl 5.8.4 or above.

Note

Using the latest Perl release is recommended. Customer Support may ask you to upgrade if you are experiencing problems.

Procedure

On the z/OS or MVS server:

- Recommended: Edit the TCPIP profile so that `JESINTERFACELEVEL` is set to `2`. If this is changed, restart the FTP server.

In the SAP CPS UI:

- Choose an existing platform process server running on UNIX or Microsoft Windows.
    - Add the JCL_FTP Job Definition Type to it.

Example

The following job definition takes a job parameter called *MESSAGE* and displays its content.

```
//*
//* A hello world program with a variable message
//*
//STEP1 EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
//SYSUT1   DD *
~MESSAGE~
/*
//SYSUT2   DD SYSOUT=*
```

The following is an example of a normal JCL batch, in this case a COBOL compiler call:

```
//*
//* Note that this job definition assumes there is a PDS named CPS.JES.OBJ,
//* to store the object code.
//*
//COMP     EXEC PGM=IGYCRCTL,REGION=4096K
//SYSPRINT DD SYSOUT=*
//SYSLIN   DD DSN=CPS.JES.OBJ(JCLTEST),DISP=(OLD)
//SYSUT1   DD UNIT=SYSDA,SPACE=(CYL,(2,2))
//SYSUT2   DD UNIT=SYSDA,SPACE=(CYL,(2,2))
//SYSUT3   DD UNIT=SYSDA,SPACE=(CYL,(2,2))
//SYSUT4   DD UNIT=SYSDA,SPACE=(CYL,(2,2))
//SYSUT5   DD UNIT=SYSDA,SPACE=(CYL,(2,2))
//SYSUT6   DD UNIT=SYSDA,SPACE=(CYL,(2,2))
//SYSUT7   DD UNIT=SYSDA,SPACE=(CYL,(2,2))
//SYSIN    DD *
       IDENTIFICATION DIVISION.
       PROGRAM-ID. JCLTEST.
       PROCEDURE DIVISION.
           STOP RUN.
/*
```

See Also

- Different Versions
- Checking Your License

Using the JCL Job Definition Type

Use

The JCL job definition type works with a *native agent*. The z/OS native agent can only be obtained from Redwood directly.

The native agent does require software to be installed on the z/OS server, but has the advantage that it requires less resources (no polling) as it uses SMF and JES exits to intercept the job at the appropriate moments. It is suited to running many concurrent jobs at high speeds with very low latency between job finish by JES and job status update in SAP CPS.

Note

You require the *ProcessServerService.JCL.limit* license key in order to be able to use the JCL job definition type. You require one unit per z/OS system.

The source of a JCL job definition contains either a complete job description consisting of JCL, or a single line naming a DSN containing the JCL on the mainframe. A JCL coding starts with a line beginning with `//`. A DSN contains just the DSN name, without any keywords.

Example of JCL in the job definition source

```
//EXAMPLEX JOB (ACCT#), 'JOB DEMO',CLASS=X
//STEP1 EXEC PROG=IEFBR14
```

Example of a DSN PDS naming the JCL on the mainframe

```
HLQ.DEMO.CNTL(JCLDEMO)
```

Parameter substitution

One of the advantages that most job definition types have is that they use native parameter substitution. The source that you type as the job definition source can use all parameters that you defined on the job definition, plus some built-in ones, in the native syntax of the computer language that you are writing in.

The syntax that SAP CPS uses for parameters is:

~parameter~

For example, if you have parameters P_DSN1 varchar2(30), P_DSN1_TRACKS number(3), you can use this in the JCL source like this:

//DSN1 DD DSN=~P_DSN1~,DISP=(NEW,CATLG),SPACE=(TRK,(~P_DSN1_TRACKS~,1))

Note

We use the tilde (~) character to start and end a parameter name, since this character was not represented in the earlier EBCDIC character sets, and thus not very likely to occur in normal JCL cards. However, current EBCDIC tables usually do have tilde, as code point A1, but it can only occur in '*' SYSIN cards, as JES does not allow the tilde character.

JOB card

Every job received by JES must start with a job card with the following format:

```
//jobname JOB (accounting),name,keyword-parameters
```

SAP CPS automatically generates a job card if it sees that you have not created one at the start of your code. This default card is:

```
//~JES_USER~~JES_JOB_IDENTIFIER~ JOB ~JES_ACCOUNT~,'~JES_USER~'[,CLASS=~JES_CLASS~][,MSGCLASS=~JES_MSGCLASS~]
```

The `,CLASS` and `,MSGCLASS` arguments are only added if they are filled by a job definition parameter.

SAP CPS communicates to z/OS using the IBM FTP server, which has a direct link to the JES internal reader and the JES spool. If the *JESINTERFACELEVEL* is 1 you must use a jobname of `~JES_USER~` followed by one character. The default job card does this correctly. If the FTP server is running at `JESINTERFACELEVEL=2` then the job name is free, and the automatic job card will use a jobcard starting with `J` followed by a

Prerequisites

The native agent will require the following z/OS configuration:

- z/OS 1.7 or higher.
- TCP/IP enabled.
- USS file system available with 20 MBytes free space.
- Willingness to install SMF exits (supplied).
- Willingness to install JES exits (supplied).

Limitations

The first release of the z/OS native agent will have the following limitations:

- The native agent does not yet support file events; it only submits and monitors jobs. Events can be sent to SAP CPS via the jevent system tool. Support for file events is expected for a later release, depending on demand.

Example

The following job definition takes a job parameter called *MESSAGE* and displays its content.

```
//*
//* A hello world program with a variable message
//*
//STEP1 EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
//SYSUT1   DD *
~MESSAGE~
/*
//SYSUT2   DD SYSOUT=*
```

The following is an example of a normal JCL batch, in this case a COBOL compiler call:

```
//*
//* Note that this job definition assumes there is a PDS named CPS.JES.OBJ,
//* to store the object code.
//*
//COMP     EXEC PGM=IGYCRCTL,REGION=4096K
//SYSPRINT DD SYSOUT=*
//SYSLIN   DD DSN=CPS.JES.OBJ(JCLTEST),DISP=(OLD)
//SYSUT1   DD UNIT=SYSDA,SPACE=(CYL,(2,2))
//SYSUT2   DD UNIT=SYSDA,SPACE=(CYL,(2,2))
//SYSUT3   DD UNIT=SYSDA,SPACE=(CYL,(2,2))
//SYSUT4   DD UNIT=SYSDA,SPACE=(CYL,(2,2))
//SYSUT5   DD UNIT=SYSDA,SPACE=(CYL,(2,2))
//SYSUT6   DD UNIT=SYSDA,SPACE=(CYL,(2,2))
//SYSUT7   DD UNIT=SYSDA,SPACE=(CYL,(2,2))
//SYSIN    DD *
       IDENTIFICATION DIVISION.
       PROGRAM-ID. JCLTEST.
```

```
        PROCEDURE DIVISION.
            STOP RUN.
/*
```

See Also

- Different Versions
- Checking Your License

IBM AS/400 Job Definition Types

Use

The following job definition type is available for AS/400 systems:

1. AS/400 Job Definition Type

Using the AS/400 Job Definition Type

Use

The AS/400 job definition type allows you to trigger jobs in remote AS/400 systems and retrieve the job output. For each remote AS/400 system you need one dedicated process server and preferably a dedicated queue.

The AS/400 job definition type allows you to start three different kinds of jobs in an AS/400 system.

- Commands (for example `DSPJOB`)
- File Actions
  - file.delete – Delete a file
  - file.create – Create a file
  - file.list – List a file or a directory
- Job Actions
  - job.waitForJobs
    - name= job name
    - user= job user

Syntax

The format of the script text is below:

Blank lines and lines starting with a # are ignored. The simplest possible job definition is simply a command:

```
command
```

If you specify a command, it can be followed by a single optional job control section, and zero or more output sections.

```
[JobControl]
Append=JOBQ(EXTERNAL) LOG (2 50 *MSG) SPLFACN(*KEEP)
```

Any parameter on the process server that starts with *AS400Job* (but not *AS400JobSpool*, these are controlled by the Output sections). To do so, specify the parameter without the *AS400Job* prefix. Names are case insensitive. For example, you can specify `Append=` to override the process server value for *AS400JobAppend*.

Note

The `SBMJOB CMD()` part of the command is added by the process server and should **not** be specified.

Output sections are optional. If you do not specify any output sections, output will be spooled to files with names based on the spool file name and number from the AS/400, and `stderr` and `stdout` will be empty. As soon as one section is specified, you have complete control over spooling. Sections are processed in the order they are specified (this is important if two output sections point to the same job file).

```
[Output]
spool=name of spool to fetch, required
number=number of spool to fetch, required, *ONLY and *LAST are accepted
append=true to append to the job file, false to overwrite it.
jobfile=Name of the job file
maxsize=Override the value of the parameter AS400JobSpoolMaxSize.
banner=Override the value of the parameter AS400JobSpoolBanner.
removecontrolcharacters=Override the value of the parameter AS400JobSpoolRemoveControlCharacters
```

Where the values override a parameter, the following order of preference is used:

- Value in [Output] section
- Job definition parameter value
- Process server parameter value
- Default value of process sever parameter
- [action] sections

Currently defined actions are:

File Related:

- file.delete - delete a file specified by filename=
- file.create - create a file specified by filename= (the file will be deleted first)
- file.list - list files in directory specified by filename=

Job Related:

- job.waitForJobs - wait for all jobs with name matching name= and user matching user= to complete. The action will wait for at least one job to exist and complete.

Any value that is valid for `SBMJOB` on your OS/400 release is valid here. The `CL` command sent to the AS/400 is:

```
SBMJOB CMD(command) AS400JobAppend
```

You can find the options for `SBMJOB` on the IBM website:

http://publib.boulder.ibm.com/infocenter/iseries/v5r4/index.jsp?topic=/cl/sbmjob.htm

You may need to specify additional value here depending on the default values on the AS/400.

Useful options are:

- LOG (2 50 `*MSG`) - change the message logging values used to determine the amount and type of information sent to the job log. This has three elements: the message (or logging) level, the message severity, and the level of message text. The values specify a logging level of 2 (messages to the job's external message queue and program message queue that exceed the severity), a severity of 50 and to write the message text to the job log (`*MSG`).
- SPLFACN(*KEEP) - Keep spooled files even after the job has ended, this allows the connector to parse the job log and determine the return code.

The customer may want to run some jobs in a different queue or subsystem. This is achieved by changing the AS/400 queue that the jobs run in (the AS/400 job then determines the subsystem). There are two ways, use `JOBQ(QUEUENAME)` in 'Submit control parameter' to change it on a per job basis, or change the default queue for the AS/400 user (the AS/400 operator can do this).

If you need to specify multiple options, separate them with a space:

```
LOG (2 50 *MSG) SPLFACN(*KEEP)
```

```
JOBQ(EXTERNAL) LOG (2 50 *MSG) SPLFACN(*KEEP)
```

How actions work

There are two types of action:

- File actions – these are primarily for diagnostics for file events, or in the case of `file.delete` for deleting flag files created by other processes.
- Job actions – there is only one job action defined – this waits for other jobs (child jobs) to complete.

File action file.delete

The action checks if the file specified by the filename= key exists. If the file does not exist then the action does nothing and the job is set to Completed. If the file exists, then the action attempts to delete it. If this succeeds the job is set to Completed, otherwise it is set to Error.

File action file.create

The action tries to create the file specified by the filename= key. If the file exists, then the action attempts to delete it. If this fails the job is set to Error. If this succeeds, then the file is opened and a single byte (0) is written to the file and the file is closed. If this succeeds then the job is set to Completed, otherwise it is set to Error.

File action file.list

The action tries to open the file or directory specified by the filename= key. If it exists, then the action attempts to list its contents. If this fails the job is set to Error. If this succeeds, then:

- If there are no files in the directory: “No files found” is printed.
- If one or more files are found in the directory, then the number of files found is printed, followed by one line for each file.
    - If the file is a directory then ‘D ‘ is printed before its name.
    - If the file is not a directory then ‘ ‘ is printed before its name.
- Errors during directory listing are written to the error file, but the listing will continue.

Job action `job.waitForJobs`

This action will wait for all jobs with name matching name= and user matching user= to complete.

The action will first wait for at least one matching one job to start, then wait for all jobs matching the criteria to reach a final state.

We use the class com.ibm.as400.access.JobList and set the name and user selection criteria. Generally these should be exact matches for the job name and user. You can find more information about possible values by browsing to http://publib.boulder.ibm.com/infocenter/iseries/v5r4/index.jsp?topic=/rzahh/javadoc/constant-values.html and searching for JobList.

Variables and Parameters

All AS/400 Process Server Parameters can be specified as job parameters with the same name to override process server defaults.

Any AS/400 Process Server parameter is automatically valid as a job definition parameter, and overrides the process server default. You don't have to create or specify these parameters on your job definitions. If they are not present the process server defaults are used. If they are present they override the process server defaults.

There are three special cases:

- The *AS400User* and *AS400Password* parameters – these should be overridden using the job definition *Run As* user.
- The *AS400Server* parameter – this can be used to specify the process server that the job will run on using a parameter rather than a queue

You can also define your own parameters, but they can't start with *AS400*, as this is reserved for the connector. These parameters can be used to substitute values inside the script text, using ${name}. Substitutions are only valid inside values, not keys (for example only after the =).

Note

When a job reaches status *Error*, the *Out* values of parameters are not always set. You can use a *Post Running* action to set them if required.

The AS400User and AS400Password job definition parameters and the Run As User

The Run As user can be used to override the default values specified by the process server parameters of the same name. The Run As user is specified as `Username/Password`. The password part of this parameter will automatically be encrypted when you save it. The encrypted value starts with {RedwoodSecureStorage}.

The AS400Server job definition parameter

The *AS400Server* job definition parameter specifies the AS/400 to run the job on. This must be the name of the AS/400 process server. If you submit an individual job and specify this parameter, then the constraints will prevent you from selecting an incorrect queue. If the job is part of a chain and you specify a queue that does not contain the process server specified by the *AS400Server* parameter, then the system will automatically reroute your job to the first queue (in alphabetical, case insensitive order) that:

- Has only the process server specified in *AS400Server* (preferred)
- Has the process server specified in *AS400Server*, and potentially other process servers.

In all cases the job will be forced to run on the process server specified by *AS400Server* regardless if other process servers on the queue can run the job.

If you want to submit a job on a queue with multiple AS/400 process servers and use the load balancing features, do not specify a value for the AS400Server job definition parameter.

Example

The following shows how to pass a variable to an AS/400 job

```
WRKOUTQ OUTQ(${outputqueue})
```

The following shows how to perform an action

```
[action]
action=file.list
filename=${filename}
```

```
DSPJOB

[JobControl]
# This AS/400 does not create job logs or keep spool files by default, so override the behavior.
Append=LOG(2 50 *MSG) SPLFACN(*KEEP)
notes=true

[Output]
# Capture QPDSPJOB (number 1) to stdout, as long as it is less than 10Mb
jobfile=stdout
spool=QPDSPJOB
number=1
maxsize=10000000

An 'Action' can be specified instead of a command.

[Action]
action=name of action
action specific parameters
 ...
```

See Also

- Different Versions
- Checking Your License

Creating Platform Process Servers

Use

Platform process servers are not available with a Free of Charge version license. They require the following keys:

- *ProcessServerService.External.limit* - the total amount of external (Platform agent and SAP) process servers
- *ProcessServerService.OS.limit* - the total amount of platform agent process servers

Platform process servers are used to perform the following activities:

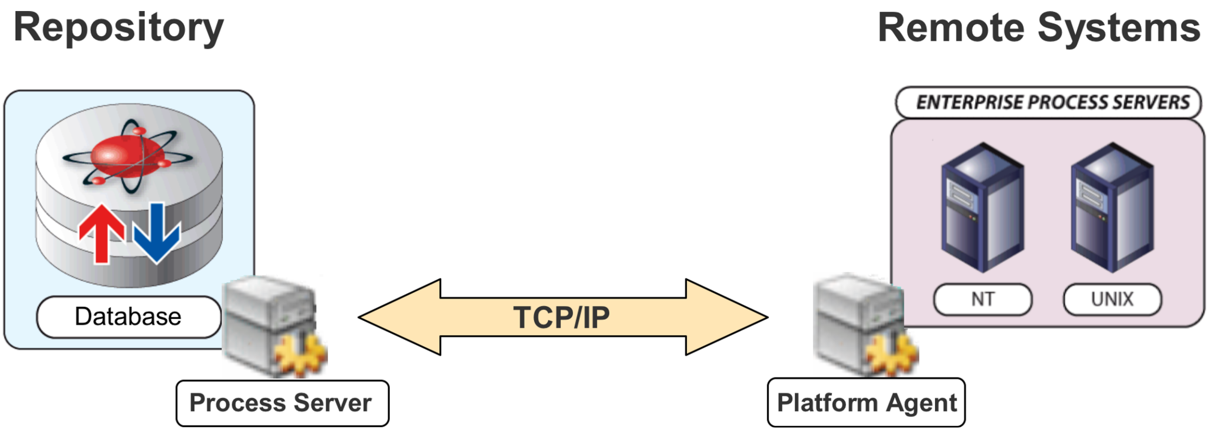
- Run Operating System jobs.
- Watch for file events.
- Observe the system load.
- Act as file servers for OS job files.

To make this possible a platform agent is installed on the computer system where you wish to run jobs. The platform agent makes the specific OS system calls required for running jobs and monitoring the system. The scheduler server communicates with the platform agent over the TCP/IP network using the HTTP or HTTPS protocols.

Note

When installing a platform agent on an SAP application server, SAP recommends to name the process server *<SID>_<APP_SVR>_PlatformAgent*, where <APP_SVR> is the hostname of the application server.

The configuration of a process server that runs OS jobs consists of settings in the Repository and the installation and configuration of the platform agent.



The process server is a logical object in the Central Server. It consists of two threads: one is responsible for send messages *to* the platform agent and the other is responsible for processing messages coming *from* the platform agent.

The process server is configured using the Process Server Parameters. Overall settings for all platform agents are configured using the Platform Agent Registry Entries.

The platform agent itself consists of a number of different executables and/or scripts, each playing their own role. How these interact is documented in more detail in the pages below, but in general there are usually at least the following programs:

- `network-processor` is the program that is responsible for communicating with the Central Server.
- (UNIX, Microsoft Windows, OpenVMS) `job-processor` is the program that is responsible for a single job. When a job starts, an instance of the `job-processor` starts.
- `jtool` and derivatives form the System Tools.

For more details see:

- Creating UNIX process servers
  - User Switching Security Options
- Creating Microsoft Windows process servers
- Creating HP OpenVMS process servers

Agent Connection

The Central Server communicates with the platform agent (the `network-processor`, to be exact) via a TCP connection carrying data in the form of the HTTP protocol. The platform agent is contacted by the scheduler central Java server, and not the other way around. In other words, in terms of the TCP/IP connection the agent is the server and the Central Server is the client. This has the advantage that no incoming TCP ports have to be enabled on the Central Server in order to schedule jobs on a machine beyond a firewall.

The default installation procedure also establishes *secrets*. These secrets are a simple form of *key* that ensure that a particular platform agent only talks to the Java server that is in possession of the same secret. Message data is also protected by a hash value. The combination of these measures means that it is very hard for an attacker to falsify the communications and act as a Java server.

If, however, you need to protect the *content* of the data connections as well, you need encryption. Encryption is done using an optional layer that is available only from Redwood directly, as it contains some third party source code. See Secure Agent Connections for more details.

Processing Platform Jobs

Use

Platform agents are designed to be robust. As such, they are divided in two parts, one running in the Central Scheduler and the other on the remote OS, named the agent; both can be started and stopped independently. The connection between the two is established when both are running and reachable over the network.

The process server in the SAP CPS server consists of one or more services, the `PlatformAgentService` service attempts to reach its platform agent defined by process server parameters. See the process server services section for more information about other available services.

The agent consists of two types of processes, the `job-processor` and the `network-processor`, which are independent as well. When the network connection fails, or you terminate the `network-processor` process, the connection to the Central Scheduler will be lost. However, jobs that were already started will continue to be processed on the OS by the `job-processor` processes until they complete. As soon as the connection to the Central Scheduler has been established again (by reconnecting the network or starting the `network-processor`), the status of the jobs is updated in the SAP CPS server. If you terminate a `job-processor` process, you terminate a job only and the network-processor will update the status of the job, which will be *Error*.

Both the `network-processor` and the `job-processor` store information on the file system and make sure the information is received by the Central Scheduler.

UNIX Process Servers

Use

The UNIX process server is implemented on UNIX by a native platform agent. The SAP CPS platform agent for UNIX is used on all supported OS platforms where the *native* API is UNIX. This means that Linux and Mac OS X are supported via the UNIX agent. Operating systems where the UNIX API is 'bolted on' as an additional layer use a different agent; see HP OpenVMS and Microsoft Windows for those.

Structure

On UNIX the platform agent is started by the `scheduler` script located in `${InstallDir}/${version}/etc`. That script starts all platform agent instances found in the various installation trees. The `scheduler` script is started at system boot time by a link from UNIX system dependent files, usually `/etc/init.d/` scripts. The `scheduler` script invokes the `platform-agent` script. You should *not* run the platform-agent script by hand, that's what the `scheduler` script is for.

Each instance is represented by at two or three continuously running processes, shown here in `ps` form:

```
/bin/sh /opt/redwood/agent/0_26_6/bin/platform-agent default
\_ /opt/redwood/agent/0_26_6/bin/network-processor -p /opt/redwood/agent/etc/pid/default/default.pid -i defau
 \_ /opt/redwood/agent/0_26_6/bin/network-processor -p /opt/redwood/agent/etc/pid/default/default.pid -i defa
```

The `platform-agent` and top level `network-processor` are present from the start, the second network processor instance is created as soon as the platform agent is configured by the process server. The platform agent is used to guard over the network processor: if it dies unexpectedly it will restart it. It also plays a role in Automatic Updating of the agent. The second instance of the network processor is used to verify that the network processor can create a child process and to guard that the OS returns proper data for child processes. All the rest of the work is done by the primary network processor, except if jobs are started. For each job a `job-processor` is started.

Note

UNIX process servers are only available when your license includes a non-zero value for the key *ProcessServerService.OS.limit*. The value of this key determines how many platform agents (across all platforms) you can run.

UNIX jobs are executed by a platform agent. It executes the code in the job definition in a specific environment as a specific user.

The platform agent may run on the same machine as the server, or on any other machine on the network. UNIX jobs contain shell scripts in languages such as KSH, BASH, CSH and Perl.

Note

When installing a platform agent on an SAP application server, SAP recommends to name the process server *<SID>_<APP_SVR>_PlatformAgent*, where <APP_SVR> is te hostname of the application server.

The installation script is named individually for each Operating System / platform combination as follows:

<OS>-<platform>.sh

The installation script can store your choices for subsequent runs, you will then be able to perform unattended installs. Your choices are saved in the installation shell script itself.

Note

The shell script is a self-extracting tar archive that starts with 'shell' script but contains a tar file at the end. When you transfer it via FTP, you must use binary mode.

The installation script accepts the following switches:

Switch	Description
-s|--store	stores settings for any subsequent installations
-f|--force	Overwrite any existing files
-i|--instance <server>:<port>	FQDN and port of the Central scheduler
-d|--destination <path>	The installation directory
-u|--unattended	Perform an unattended installation
-l|--language	Set the language of the installation
-a|--enable-update <Y/N>	Enables/disables automatic update of the platform agent
-c|--clear	Clear out any values from previous installs
-rs|--registration-server	FQDN and port of the central SAP CPS server, used for platform-agent registration
-un|--username	username, needs to have the `scheduler-administrator` role, used for platform agent registration
-pw|--password	users' password, used for platform agent registration
-ps|--process-server	Desired name of the process server in SAP CPS, used for platform agent registration
-ql|--queue-list	Comma-separated list of queues for the process server, used for platform agent registration
-h|--help	Prints the syntax

Valid languages are: *EN*, *NL*, *FR*, and *DE*.

Note

To be able to use the unattended installation you must have used this same script on another system of the same platform/OS combination with the `s` switch, which saves the values you enter for successive installations.

Use the switches as follows:

<OS>-<platform>.sh <switch>

Example

The GNU/Linux installation script for x86 platforms, used with switches to perform an unattended installation in German.

```
$ linux-x86.sh -u -l DE
```

Platform independent configuration instructions can be found in the Configuring Platform Agents section.

Prerequisites

- A running SAP CPS system
- A UNIX server with appropriate access to the server (including root password). The following Operating Systems are supported:
  - AIX 5.1 or higher (Power 5 32/64-bit)
  - HP-UX 11.00 or higher (IA64 32/64-bit)
  - HP-UX 11.00 or higher (PA-RISC 32/64-bit)
  - Linux (IA64 64-bit) with glibc 2.4 or higher
  - Linux (Power 64-bit) with glibc 2.4 or higher
  - Linux (x86 32/64-bit) with glibc 2.4 or higher
  - Mac OS X 10.4 or higher (x86 32-bit)
  - Solaris 8 or higher (Sparc 32/64-bit)
  - Solaris 10 or higher (x86_64 64-bit)
  - Tru64 UNIX 5.0 or higher (Alpha)
- An UTF-8 character set is required, for example `en_US.UTF8`
- File system and file data character sets are assumed to be `ISO8859-15`, you can overrule them by specifying the --filesys and --filedata command line options.
- The Korn shell, ksh or pdksh, needs to be installed if you are going to run KSH jobs. The shell is available by default on most UNIX platforms, and is often the default shell. Not all Linux have ksh installed by default. There is an open source implementation usually known as `pdksh` that can be installed on most distributions.
  - The ksh binary must reside in `/bin/ksh` or you will have to create a symbolic link called `/bin/ksh` pointing at it.
  - On Linux, the `zsh` shell also provides `/bin/ksh` but its implementation is not POSIX compatible and may not be used.
- A Security Option. When you install a UNIX platform agent that will run jobs as multiple users, you will need to decide what security option you would like to use. SAP recommends you read the Choosing an User Switching Security Option through and discuss it with your system, network and security administration teams before installing.

Note

If you are running Linux and are unsure of the installed glibc version, you can execute /lib/libc.so.6 in a terminal.

Note

You must install the SAP CPS platform agent on a local file system; SAN file systems might be considered local, when they are mounted as iSCSI, for example.

You will encounter the following error message, when `ksh` is not installed on your system in `/bin/ksh` in the log of a *KSH* job:

```
error 2008-09-19 12:33:30,811 WST [14188-Main tid=4188101184] opsys.job
- Cannot execute Redwood Job 3281 '/home/user/jobs/3281/run_KSH': No such file or directory
```

Procedure

The procedure consists of two steps:

- Installing the platform agent on the machine that needs to run jobs.
- Configuring the process server in the repository.

SAP recommends that you install and start the platform agent on the machine that you want to run jobs first, and then configure the process server, however you can do these steps in any order.

Pre-Installation Tasks

Create an account (optional)

SAP recommends that you create a new account and group to contain the platform agent that satisfies your local naming requirements.

This documentation uses the account *redwood* and group *redwood*.

1. Create the account: `groupadd redwood`.
2. Create the account: `useradd -g redwood -s /bin/ksh redwood`.
3. Create the home directory of the user: `mkdir /home/redwood`.
4. Change ownership of the directories: `chown redwood:redwood /home/redwood`.
5. Set the password for the account: `passwd redwood`.
6. Login as redwood: `su - redwood` or `login redwood`.

Download the Software

1. Navigate to *Configuration* > *Software*.
2. Download the appropriate file for the platform you want to run jobs on and copy it to your server.

Installation

1. Run the installation shell script for your platform
2. Choose the language for the installation
3. Choose an installation location, like `/home/redwood/`, hit *y* to create the directory.

4. Choose an instance name.
5. Choose a port the platform agent will use.
6. Choose the security option
7. Choose whether you want to register a process server with the server; this option allows you to create a process server on the central server and connect it to the agent.
   a. If you chose to register the process server with the server, enter the FQDN of the central server, otherwise, continue at step 8.
   b. Choose the port number of the central server
   c. Choose a name for the process server
   d. Choose if you want to lock the agent to the central server, this prevents other central servers from using this platform agent instance.
   e. Enter a username for the central server, the user must have administrative privileges
   f. Enter a password for the user
8. Choose if the platform agent should run at system startup, requires `SysV init` startup script support
9. Choose if you want to update the platform agent automatically, when the central server is updated. The central server will check the version at every startup and if they differ, the central server will push the new platform agent software to the host and have it installed automatically.
10. Run the *root.sh* as user `root`

Further platform independent configuration instructions can be found in the Configuring Platform Agents section.

Starting the platform agent at startup

Starting the platform agent at system startup is recommended. How this is done largely depends on your platform. If your platform supports SysV init, then the installer should do it for you once you ran the *root.sh* as user `root`.

If the installer was unable to configure automatic startup, you need to include the command corresponding to your security option in your system startup scripts.

Creating the Process Server

The installer creates a Process server automatically by contacting the server at the provided address during the installation procedure. If this fails or you want to do this yourself, follow the following steps.

1. Navigate to *Environment* > *Process Servers*.
2. Choose Create from the context menu.
3. Fill in the values below

Automatic updating

The UNIX platform agent supports Automatic Updating.

If the network processor is started by the platform agent the agent is capable of being [updated automatically|Cronacle8DocAgentAutoUpdate]. The administrator can control whether auto-update should be performed by creating or removing the `${InstallDir}/etc/startup/${instance}/autoupdate` directory.

Removing the software

There is no automated process for removing the software. However, the following will remove everything associated with the software:

1. Find the install directory of the software you want to remove, this is ${InstallDir}
2. Stop all instances running in this directory.
3. Remove the installation directory ${InstallDir}.
4. Check `/etc/init.d/`, `/etc/rc.d/init.d`, `/sbin/init.d`, `/sbin/rc3.d` or `/Library/LaunchDaemons` (depending on the UNIX variety). If it contains a `scheduler` script, edit it and remove the ${InstallDir} path from the line in the beginning of the scripts containing `PATHS=`. If you have no remaining installations of the platform agent on this machine, remove the script entirely.

Values

The necessary configuration Data required to configure a Unix process server is available in the UNIX Process Server Configuration Data section of the documentation.

UNIX Process Server Configuration Data

Use

The following configuration data is required to successfully configure a Unix process server.

Process Server Tab

Name	Value
Name	Alphanumeric and underscores
Default Job Time Zone	Olson time zone. Type any letter combination in the name to restrict the list, for example `lin` for *Europe/Berlin*

Job Definition Types Tab

Job definition types are required to execute job definitions of a corresponding type.

Field	Value
Job Definition Type:	*KSH*
Job Definition Type:	*JobChain*

Parameters Tab

The repository needs to know the host name (or IP address) and port number of the platform agent. The default port is 1555 and should be kept that way, unless another installed program is already using the default port.

Name	Value
RemoteHostName	<host name or IP address of system running platform agent>
SharedSecret	The shared secret as generated in the last step of the installation

All other parameters are optional, you may specify them, but they have usable defaults. See process server parameters for a list of parameters. If you do not know the value of the *SharedSecret*, see the Configuring Platform Agents section.

Queue Providers Tab

The queues the process server will serve, that is execute jobs from, are listed here.

Field	**Value**
Queue	*System* (or your own defined queue)

Resources Tab

Resources link job definitions to two or more process servers to spread the load. Process servers which share a same resource are the only ones that can run job definitions which have that resource. You may choose any name for a resource. Jobs that require a specific resource are spread across all process servers that have that resource and serve the queue the job was submitted in.

Services Tab

Services are used to execute jobs. When you specify a job definition type, the corresponding service is added automatically. You only need to fill in this tab manually if you want to create a file event only process server.

Choosing a User Switching Security Mode

Use

On UNIX platforms, it is common to have a separate user for an application. When you need to interact with these applications, commands need to be executed as a specific user.

There are several ways to switch users on UNIX systems, and the scheduler supports these.

Note

In all the scenarios we assume that the software is installed as user 'redwood'.

The installation script of the platform agent requires you to choose a security option from the following:

1. Plain: Running jobs only as a normal user.
2. Root: Running jobs as any user, the platform agent running as root (Not recommended).
3. Setuid: Running jobs as any user, the platform agent running as an unprivileged user, using setuid.
4. Sudo: Running jobs as any user, the platform agent running as an unprivileged user, using sudo.

If you want the scheduler to be able to run jobs using a variety of user accounts the best options are *setuid* and *sudo*. The *setuid* method works on all UNIX systems, and does not require additional software. It does mean that one binary (the job-processor) is installed with elevated privileges. Some sites do not allow this, and therefore the *sudo* mechanism is supported as well. In the *setuid* case the list of allowed users is configured in the scheduler configuration files only. In the *sudo* case the configuration of the `sudo` command is also taken into account. For users that do not require the additional level of security that *sudo* offers, we recommend that the *setuid* scenario is used as it does not require the installation of additional software (if `sudo` is not yet installed.)

The installation script will ask for the scenario you have chosen. It will store the requested user switching mode in a configuration file. The network-processor will then use the scenario defined there. To change this configuration setting afterwards, see Configuring Platform Agents.

The following sections describe the different scenarios in detail,

1. Plain: Running jobs only as a normal user.

In this scenario, no user switching is done. All jobs will run using the same UNIX account. Jobs that specify that they need to run under a particular user account will fail.

No special action needs to be performed to prepare this scenario.

The startup script will switch to the user specified at installation time and run the `network-processor` under that account.

2. Root: Running jobs as any user, the platform agent running as root

In this scenario, user switching is done. You can run jobs as any user. You can limit the UNIX accounts the jobs run under by creating a blacklist or whitelist.

The startup script will switch to the root user and run the `network-processor` under that account.

3. Setuid: Running jobs as any user, the platform agent running as an unprivileged user, using setuid

In this scenario, user switching is done. You can run jobs as any user. UNIX level user switching is done using the setuid bit on the job-processor. Changing file ownership is also done using the setuid bit.

This scenario is easy to set up, as it does not require the possible installation of `sudo`, and executes quickly. You can limit the UNIX accounts the jobs run under by creating a blacklist or whitelist.

The startup script will switch to the non-root user specified at installation time and run the `network-processor` under that account. When a job runs, the `job-processor` will change the ownership of the job files to the correct account, run the job and then change the file ownership back to the scheduler user.

4. Sudo: Running jobs as any user, the platform agent running as an unprivileged user, using sudo

In this scenario, user switching is done. You can run jobs as any user. UNIX level user switching and changing file ownership is done using the `sudo` command.

This requires that you have `sudo` installed and configured.

This scenario allows you to limit the control of switch user accounts to the sudo control files (usually /etc/sudoers). The only thing that runs as root is the `chown` command for files that are under the control of the scheduler.

- The startup script will switch to the non-root user specified at installation time and run the `network-processor` under that account.
- When a job runs, first `sudo -S -u root chown` will be called to change the ownership of the job files to the correct account.
- The job is run via `sudo -S -u <user> job-processor`.
- When the job-processor has finished a final `sudo -S -u root chown` will change the ownership of the job files back to the user running the network-processor.

This scenario does take a little more effort to configure, as you may have to install `sudo` or a comparable security system, and configure that security system so the platform agent can do what it needs to do.

The installer script will configure /etc/sudoers (or prepare for it when not run as root). If you use a different solution than sudo, run the installer as follows:

```
$ VISUDO=your-visudo-equivalent $SHELL ./installer.sh
```

If your security solution does not provide a `visudo` compatible program, just substitute `vi`.

After the installation finishes, remove the sudo commands in the root.sh file and configure your solution manually. You may also need to override the *SwitchUserCommand* parameter on the process server.

Note

The *SwitchUserCommand* process server parameter is only used for `sudo` User Switch Mode and should not be set.

Security Considerations

When you use sudo user-switching mode, a user who gained access to the SAP CPS user could access files owned by `root`. To avoid this, you can specify your own sudo command. SAP CPS ships with an example `sudo.sh` shell script which checks the parameters and attempts to improve security. You may edit this file to match your security requirements. See Configuring Platform Agents for more information.

Creating a Microsoft Windows Process Server

Use

Note

Microsoft Windows process servers are only available when your license includes a non-zero value for the key 'ProcessServerService.OS.limit'. The value of this key determines how many platform agents (across all platforms) you can run.

Microsoft Windows jobs are executed by a platform agent that is installed on the Microsoft Windows server where you want the job executed. It executes the code in the job definition in a specific environment as a specific user. The platform agent may run on the same machine as the SAP CPS server, or on any other machine on the network.

The installer unpacks the software, sets up a service on the local system and registers it with a SAP CPS server. Once the installer is finished, you will have a running platform agent service on the target machine and a process server in the SAP CPS server, if you registered it. You can choose not to register the platform agent with the Central Scheduler. However, you will have to manually setup a process server in SAP CPS and configure it to connect to the platform agent.

Platform-independent configuration instructions can be found in the Configuring Platform Agents section.

The platform agent is compatible with Microsoft Cluster Service, and can be configured on cluster nodes to be highly available. More details can be found in

the Configuring Platform Agents on Microsoft Cluster Nodes section.

Note

You must install the SAP CPS platform agent on a local file system; SAN file systems might be considered local, when they are mounted as iSCSI, for example.

Structure

On Microsoft Windows the platform agent is automatically started by the Windows Service Controller. It starts a service with display name *Scheduler Platform Agent ${instance} ${version}* and service name `jcs_${version}_${instance}` . The service executable, which you will find in tools such as the *Task Manager*, is `network-processor.exe`. There is no intermediary program called `platform-agent` as on UNIX; the role of the `platform-agent` script which is monitoring that the `network-processor.exe` does not die undetected is performed by the Microsoft Service Controller.

For each job a `job-processor.exe` is started. This will run under the account set via the *Run As* credentials.

Installer Usage

The Windows platform agent installer accepts the following arguments, if you specify `-u`, you have to specify all the others as well for unattended installation.

```
windows-x86.exe  [[-u|g filename] [[-d destination] [[-i name] [[-l lang] [[-rs server:port] [[-un username]
```

Argument	Description
u	Unattended mode, logs to filename
g	GUI mode, logs to filename
d	Destination (installation path)
i	Instance name
l	Language abbreviation ("en")
rs	FQDN and port of the central SAP CPS server, used for platform agent registration
un	SAP CPS username, needs to have the `scheduler-administrator` role, used for platform agent registration
pw	SAP CPS users' password, used for platform agent registration
ps	Desired name of the process server in SAP CPS, used for platform agent registration
ql	Comma-separated list of queues for the process server, used for platform agent registration

Note

The platform agent must be installed by a user who has the right to create services. SAP recommends using a user who has local administrator privileges.

Prerequisites

- Windows XP, 2003, Vista or 2008 with the latest Service Pack.
- Physical or remote (Terminal Services, vnc, pcanywhere) access to the system
- Local administrative privileges on the Microsoft Windows server to create a service.
- Administrative privileges in SAP CPS to register the platform agent or create the process server.

Procedure

Install the platform agent

1. Navigate to *Configuration* > *Software*.
2. Download the `windows-x86` executable and copy it to the appropriate machine (if necessary).
3. Run that executable on the Microsoft Windows system.
4. Select the language to use for installation.
5. Select an installation directory. This will be referred to as ${InstallDir} in the rest of this document.
6. Enter a *Preferred instance name*, a *Port*, whether to enable *auto-update* and optionally an Account. Enabling auto-update and always running the platform agent using the system account is recommended. Choose *OK*.
7. Optionally (but strongly recommended, except for cluster environments) you can register the platform agent by creating a new process server. Enter the FQDN (Fully Qualified Domain Name) of the Central Scheduler server in the *Server* field and its port. You can enter a name for the process server which will be used throughout SAP CPS. The default name is the instance name. You can also specify a list of queues that this process server will be serving (comma-separated, without spaces). Choose *OK*.
8. You will now be asked to log on to the Central Scheduler so the installer can create and configure your process server inside SAP CPS, this step must be skipped when you are installing on a cluster. You must use an account which has the *scheduler-administrator* role or equivalent privileges in SAP CPS.

Further configuration instructions can be found in the Configuring Remote Agents section.

Configuring a Firewall

If you run a firewall, add a rule to allow incoming connections on the port that you defined during the installation for program `${InstallDir}/bin/network-processor.exe` . As the network-processor runs as a service, your firewall program may not show a pop-up that allows you to add this rule interactively. If you are running a firewall and no such notice popped-up during installation (when the service was started), add the rule manually.

1. Navigate to *Start* > *Control Panel* > *Network Connections*.
2. Choose "Properties" from the context menu of the desired adapter.
3. Choose the *Advanced* tab.
4. Choose *Settings*.
5. Choose the *Exceptions* tab.
   a. Add either a program `${InstallDir}/bin/network-processor.exe` or port `1555`.
6. Choose *OK* twice.

Creating the Process Server

This step is required when you install on a cluster. You should not do this if you registered the process server during installation.

- Navigate to *Environment* > *Process Servers*.

- Choose Create from the context menu.
- See Configuration of a Microsoft Windows Process Server for configuration data.

Automatic updating

The Microsoft Windows platform agent supports Automatic Updating.

If the service runs as the *Local System* account the agent is capable of being updated automatically. If the administrator needs to change the setting at a later date you can do this as follows:

- To disable auto-update remove the `${InstallDir}\etc\startup\${instance}\autoupdate` directory.
- To enable auto-update create the `${InstallDir}\etc\startup\${instance}\autoupdate` directory and change the service so it runs as *Local System* by starting the Microsoft Windows Service Manager (`services.msc`) and modifying the *Log On* dialog of the Scheduler Platform Agent service.

Removing the service and software

There is no automated process for removing the service and software. However, the following will remove everything associated with the service and software:

1. Find the install directory of the software you want to remove, this is ${InstallDir}
2. Use the Redwood Service Manager at ${InstallDir}/bin/servicemanager.exe to stop and remove all running services.
3. Remove the installation directory ${InstallDir}.

Configuration of a Microsoft Windows Process Server

Use

The installer automatically creates a process server for you in SAP CPS, you can customize the process server to fulfill your needs.

Process Server Tab

Name	Value
Name	the name of the process server, Alphanumerical characters and underscores _ are allowed
Default Job Time Zone	Olson time zone

Job Definition Types Tab

Field	Value
Job Definition Type:	*CMD*

Parameters Tab

The repository needs to know the host name (or IP address) and port number of the platform agent. The default port is 1555 and should be kept that way, unless another installed program is already using the default port.

Name	Value

RemoteHostName	<host name or IP address of system running remote agent>
SharedSecret	<the shared secret generated during the installation>

Note

On Windows systems, the shared secret is generated during installation and gets written to the following file on the remote agent `${InstallDir}\net\instance\<instance>\private\secret` . The default instance is `default`.

All other parameters are optional, you may specify them, but they have usable defaults. See process server parameters for a list of parameters. If you do not know the value of the *SharedSecret*, see the Configuring Remote Agents section.

Note

To be in compliance with Microsoft guidelines, you should define the *DataRootDirectory* process server parameter and set it to a location outside of the `Program Files` directory.

Queue Providers Tab

Field	**Value**
Queue	*System* (or your own defined queue)

Resources Tab

Resources link job definitions to two or more process servers to spread the load. Process servers which share a same resource are the only ones that can run job definitions which have that resource. You may choose any name for a resource. Jobs that require a specific resource are spread across all process servers that have that resource and serve the queue the job was submitted in.

Services Tab

The correct services will be added for you automatically, you only need to fill in this tab manually if you want to create a file event only process server.

Creating HP OpenVMS Process Servers

Use

Note

OpenVMS process servers are only available when your license includes a non-zero value for the key *ProcessServerService.OS.limit* **and** a non-zero value for the key *ProcessServerService.VMS.limit*.

VMS jobs are executed by a platform agent. It executes the code in the job definition in a specific environment as a specific user.

The platform agent may run on the same machine as the server, or on any other machine on the network. VMS jobs contain DCL script source: only job definitions that specify the DCL job definition type can run on OpenVMS platform agents (and only there).

The installation script is named individually for each Operating System / platform combination as follows: `openvms-<cpu>.zip`.

Platform independent configuration instructions can be found in the Configuring Platform Agents section.

Prerequisites

- A running SAP CPS system
- An HP OpenVMS server with appropriate access to the server (including root password). The following Operating Systems are supported:
  - HP OpenVMS 7.3 or higher (Alpha)
  - HP OpenVMS 8.2 or higher (ia64)
- The VMS Internationalization Kit must be installed.
- An `UNZIP` program must be installed.
- An OpenVMS account with (at least) the following VMS system privileges:
  - TMPMBX to create temporary mailboxes for communication between processes.
  - NETMBX to create a network socket for the server to communicate with the agent.
  - Optional: IMPERSONATE to run jobs as other VMS users.
  - Optional: CMKRNL to monitor the system load (page rate, CPU usage).
  - Optional: SYSPRV privilege to add a new nametable to LNM$SYSTEM_DIRECTORY
  - Optional: BYPASS privilege to manipulate job files created by jobs running as other users and to create a new nametable in LNM$SYSTEM_DIRECTORY.

Procedure

Installing the "VMS Internationalization Kit" Layered Product

The SAP CPS platform agent communicates with the Central Scheduler Server using the UTF-8 character set. To convert this to/from the usual DEC International Character Set (of which ISO-8859-1 is a superset) you need additional Internationalization Data.

If you are unsure whether the VMSI18N kit has already been installed, you can verify this by typing the following command:

```
$ product show prod *i18*
----------------------------------- ----------- ------------
PRODUCT                             KIT TYPE    STATE
----------------------------------- ----------- ------------
DEC AXPVMS VMSI18N V8.2             Full LP     Installed
----------------------------------- ----------- ------------

1 item found
```

You can also wait until you try to run one of the images supplied as part of the SAP CPS platform agent kit; all images attempt to access the necessary APIs for character set conversion when they start. If you run it on a system that does not have the kit installed, you will see an error message such as this immediately upon starting the image:

```
$ run jsleep.exe
```

```
error 2008-01-01 00:00:00,000 GMT [154-Main tid=2075815240] opsys.conv - Unable to determine proper UTF-8 cha
error 2008-01-01 00:00:00,000 GMT [154-Main tid=2075815240] opsys.conv - Make sure the VMSI18N kit from the O
```

To install the OpenVMS Internationalization Data Kit follow the instructions in the OpenVMS Upgrade and Installation manual. In short:

- Mount the "Operating System" DVD or the "Layered Products" CD of the OpenVMS media kit
- run `PRODUCT INSTALL VMSI18N /SOURCE=dev:[dir]`

Exact CD/DVD locations in our media kits are:

Release	Processor	Media label	Directory
8.2	Alpha	ALP082LP	[VMSI18N_ALPHA082.KIT]
8.3	Alpha	ALP083LP	[VMSI18N_ALPHA083.KIT]
8.3	ia64	I64082	[VMSI18N_I64083.KIT]
8.3	ia64	I64083	[VMSI18N_I64083.KIT]

You can find more information in the OpenVMS documentation:

- OpenVMS 8.2: h71000.www7.hp.com/doc/82FINAL/ba322-90032/00/01/105-con.html [http://h71000.www7.hp.com/doc/82FINAL/ba322-90032/00/01/105-con.html]
- OpenVMS 8.3: docs.hp.com/en/BA322-90077/apf.html [http://docs.hp.com/en/BA322-90077/apf.html]

Until you install the internationalization kit you can also run the software by telling it that all files are already in UTF-8. This will mangle content that uses codepoints that are not at the same location as in UTF-8, but these seldom occur in OpenVMS DCL files. You can tell the software to assume everything is in UTF-8 by setting a logical LANG to UTF-8, or add the following command line arguments: `--filesys UTF-8 --filedata UTF-8 .`

Create an account

We recommend that you create a new account to contain the platform agent that satisfies your local naming requirements.

This documentation uses the account `REDWOOD` with UIC [377,200] as an example.

To add the account

1. Log in as `SYSTEM`
2. Change to the system directory and start the authorization image: `$ MC AUTHORIZE`
3. Create the account: `ADD REDWOOD /UIC=[377,200]`.
4. Set password and login flags: `MODI REDWOOD /PASSWORD=XXXX /FLAGS=NODISUSER`.
5. Set the account privileges: `MODI REDWOOD /PRIV=(NETMBX,TMPMBX,IMPERSONATE,CMKRNL,SYSPRV,BYPASS)`.
6. Set the account default privileges: `MODI REDWOOD /DEFPRIV=(NETMBX,TMPMBX,IMPERSONATE,CMKRNL,SYSPRV,BYPASS)`.
7. Set the home directory: `MODI REDWOOD /DEVICE=SYS$USER /DIR=[REDWOOD]`.

8. Set the proper quotas for the new account: `MODI REDWOOD/JTQUOTA=0/MAXDETACH=0/MAXJOBS=0/BYTLM=0`

9. Set the quota to (at least) these values:

   a. `MODI REDWOOD/BIOLM=150/DIOLM=150/ENQLM=2000`

   b. `MODI REDWOOD/WSDEF=4096/WSQUO=32768/WSEXTENT=65536/PGFLQUOTA=100000`

10. Exit authorize: `EXIT`

11. Create the home directory: `$ CREATE/DIR SYS$USER:[REDWOOD] /OWNER=REDWOOD`.

If running the redwood account with reduced privileges:

1. If you have not granted the `SYSPRV` (or write access to LNM$SYSTEM_TABLE) and `BYPASS` privileges, create a name table: `CREATE/NAME/ATTR=NO_ALIAS/PARENT_TABLE=LNM$SYSTEM_DIRECTORY/EXE/PROT=(O:RWD,S:RWD,G:RW,W:RW) LNM$SCHEDULER`.

Installation

At the moment there is no full installation procedure yet. We suggest that you follow the following steps to make the installation compatible with how a future installation procedure will function.

1. Install 'unzip', either from the OpenVMS Freeware CD-kit, or download it from www.openvms.compaq.com/openvms/freeware/freeware.html [http://www.openvms.compaq.com/openvms/freeware/freeware.html]

2. Log in using the designated account.

3. Create a directory (preferably "AGENT").

4. Unpack the zip file to this directory.

```
$ unzip openvms-<CPU>.zip
```

The `[AGENT]` directory is called the *InstallDir* and the [BIN] directory is called the *ExecutableDir*.

Creating symbols

We recommend that you create symbols for the jtool programs so that they can be executed from within jobs, for instance in a procedure that is set up

in `SYLOGIN.COM`. We recommend you add the following:

```
$ !
$ ! Example, replace SYS$USER with your own logical or path to redwood directory.
$ !
$ !
$ JTOOL=="$SYS$USER:[REDWOOD.AGENT.<version>.BIN]JTOOL.EXE TOOL"
$ JLINK=="$SYS$USER:[REDWOOD.AGENT.<version>.BIN]JTOOL.EXE LINK"
$ JSLEEP=="$SYS$USER:[REDWOOD.AGENT.<version>.BIN]JTOOL.EXE SLEEP"
$ JREGISTER=="$SYS$USER:[REDWOOD.AGENT.<version>.BIN]JTOOL.EXE REGISTER"
$ JEVENT=="$SYS$USER:[REDWOOD.AGENT.<version>.BIN]JTOOL.EXE EVENT"
$ JMESSAGE=="$SYS$USER:[REDWOOD.AGENT.<version>.BIN]JTOOL.EXE MESSAGE"
$ JDESCRIPTION=="$SYS$USER:[REDWOOD.AGENT.<version>.BIN]JTOOL.EXE DESCRIPTION"
$ JSECRET=="$SYS$USER:[REDWOOD.AGENT.<version>.BIN]JTOOL.EXE SECRET"
$ !
$ ! In case you need to pass arguments, e.g. non-default instance
$ !
$ NETWORKPROCESSOR=="$SYS$USER:[REDWOOD.AGENT.<version>.BIN]NETWORK-PROCESSOR.EXE"
```

Further platform independent configuration instructions can be found in the Configuring Platform Agents section. In particular, if you want to guard the platform-agent against intruders and use by different server instances we suggest that you create a *SharedSecret* using the instructions found there.

Starting the platform agent interactively

For debugging and initial testing purposes it can be useful to start the network-processor interactively. For instance, to enable debugging until the process server is configured, run the following whilst logged in as *redwood*:

```
$ DEFINE RW_LOGLEVEL DEBUG
$ RUN [.AGENT.<version>.BIN]NETWORK-PROCESSOR
```

Starting the platform agent at startup

Starting the platform agent at system startup is recommended. For OpenVMS, we suggest starting with a single instance ('Default') running on port 1555. To do this, simply create a `STARTUP_DEFAULT.COM` script that you call from `SYSTARTUP_VMS.COM`. Suggested content is:

```
$ RUN/UIC=[REDWOOD] SYS$USER:[REDWOOD.AGENT.<version>.BIN]NETWORK-PROCESSOR.EXE /OUTPUT=.... /PROCESS_NAME=NW
```

Creating the Process Server

Once the platform agent (network-processor.exe) is running you can go to the scheduler UI and configure a process server that will contact the platform agent.

1. Navigate to *Environment* > *Process Servers*.

2. Choose Create from the context menu.

3. Fill in the values below

Values

Information on the required values for the creation of HP OpenVMS process servers can be found in the Configuration Data for Creating HP OpenVMS Process Servers section of the documentation.

Installing the Platform Agent on HP OpenVMS

Use

Note

OpenVMS process servers are only available when your license includes a non-zero value for the key *ProcessServerService.VMS.limit*. Process servers that include the DCL job definition type will not be counted towards the *ProcessServerService.OS.limit* and *ProcessServerService.External.limit* license keys.

Prerequisites

- A running SAP CPS system

- An HP OpenVMS server with appropriate access to the server (including root password). The following Operating Systems are supported:
  - HP OpenVMS 7.3 or higher (Alpha)
  - HP OpenVMS 8.2 or higher (ia64)
- The VMS Internationalization Kit must be installed.
- An `UNZIP` program must be installed.
- An OpenVMS account with (at least) the following VMS system privileges:
  - TMPMBX to create temporary mailboxes for communication between processes.
  - NETMBX to create a network socket for the server to communicate with the agent.
  - Optional: IMPERSONATE to run jobs as other VMS users.
  - Optional: CMKRNL to monitor the system load (page rate, CPU usage).
  - Optional: SYSPRV privilege to add a new nametable to LNM$SYSTEM_DIRECTORY
  - Optional: BYPASS privilege to manipulate job files created by jobs running as other users and to create a new nametable in LNM$SYSTEM_DIRECTORY.

Procedure

Installing the "VMS Internationalization Kit" Layered Product

The SAP CPS platform agent communicates with the Central Scheduler Server using the UTF-8 character set. To convert this to/from the usual DEC International Character Set (of which ISO-8859-1 is a superset) you need additional Internationalization Data.

If you are unsure whether the VMSI18N kit has already been installed, you can verify this by typing the following command:

```
$ product show prod *i18*
----------------------------------- ----------- ------------
PRODUCT                             KIT TYPE    STATE
----------------------------------- ----------- ------------
DEC AXPVMS VMSI18N V8.2             Full LP     Installed
----------------------------------- ----------- ------------

1 item found
```

You can also wait until you try to run one of the images supplied as part of the SAP CPS platform agent kit; all images attempt to access the necessary API's for character set conversion when they start. If you run it on a system that does not have the kit installed, you will see an error message such as this immediately upon starting the image:

```
$ run jsleep.exe
error 2010-01-01 00:00:00,000 GMT [154-Main tid=2075815240] opsys.conv - Unable to determine proper UTF-8 cha
error 2010-01-01 00:00:00,000 GMT [154-Main tid=2075815240] opsys.conv - Make sure the VMSI18N kit from the O
```

To install the OpenVMS Internationalization Data Kit follow the instructions in the OpenVMS Upgrade and Installation manual. In short:

- Mount the "Operating System" DVD or the "Layered Products" CD of the OpenVMS media kit
- run `PRODUCT INSTALL VMSI18N /SOURCE=dev:[dir]`

Exact CD/DVD locations in our media kits are:

Release	Processor	Media label	Directory
8.2	Alpha	ALP082LP	[VMSI18N_ALPHA082.KIT]
8.3	Alpha	ALP083LP	[VMSI18N_ALPHA083.KIT]
8.3	ia64	I64082	[VMSI18N_I64083.KIT]
8.3	ia64	I64083	[VMSI18N_I64083.KIT]

You can find more information in the OpenVMS documentation:

- OpenVMS 8.2: h71000.www7.hp.com/doc/82FINAL/ba322-90032/00/01/105-con.html [http://h71000.www7.hp.com/doc/82FINAL/ba322-90032/00/01/105-con.html]
- OpenVMS 8.3: docs.hp.com/en/BA322-90077/apf.html [http://docs.hp.com/en/BA322-90077/apf.html]

Until you install the internationalization kit you can also run the software, but you will have to set the characterset to UTF-8. This will mangle content that uses codepoints that are not at the same location as in UTF-8, but these seldom occur in OpenVMS DCL files. You can set the characterset of the software to UTF-8 by setting a logical LANG to UTF-8, or add the following command line arguments: `--filesys UTF-8 --filedata UTF-8` . Although this should work, it is not recommended by Redwood.

Create an account

We recommend that you create a new account to contain the platform agent that satisfies your local naming requirements.

This documentation uses the account `REDWOOD` with UIC [377,200] as an example.

To add the account

1. Log in as `SYSTEM`
2. Change to the system directory and start the authorization image: `$ MC AUTHORIZE`
3. Create the account: `ADD REDWOOD /UIC=[377,200]`.
4. Set password and login flags: `MODI REDWOOD /PASSWORD=XXXX /FLAGS=NODISUSER`.
5. Set the account privileges: `MODI REDWOOD /PRIV=(NETMBX,TMPMBX,IMPERSONATE,CMKRNL,SYSPRV,BYPASS)`.
6. Set the account default privileges: `MODI REDWOOD /DEFPRIV=(NETMBX,TMPMBX,IMPERSONATE,CMKRNL,SYSPRV,BYPASS)`.
7. Set the home directory: `MODI REDWOOD /DEVICE=SYS$USER /DIR=[REDWOOD]`.
8. Set the proper quotas for the new account: `MODI REDWOOD/JTQUOTA=0/MAXDETACH=0/MAXJOBS=0/BYTLM=0`
9. Set the quota to (at least) these values:
   a. `MODI REDWOOD/BIOLM=150/DIOLM=150/ENQLM=2000`

b. `MODI REDWOOD/WSDEF=4096/WSQUO=32768/WSEXTENT=65536/PGFLQUOTA=100000`

10. Exit authorize: `EXIT`

11. Create the home directory: `$ CREATE/DIR SYS$USER:[REDWOOD] /OWNER=REDWOOD`.

If running the redwood account with reduced privileges:

1. If you have not granted the `SYSPRV` (or write access to LNM$SYSTEM_TABLE) and `BYPASS` privileges, create a name table: `CREATE/NAME/ATTR=NO_ALIAS/PARENT_TABLE=LNM$SYSTEM_DIRECTORY/EXE/PROT=(O:RWD,S:RWD,G:RW,W:RW) LNM$SCHEDULER`.

Installing the Platform Agent Software

1. Install 'unzip', either from the OpenVMS Freeware CD-kit, or download it from www.openvms.compaq.com/openvms/freeware/freeware.html [http://www.openvms.compaq.com/openvms/freeware/freeware.html]

2. Log in using the designated account.

3. Create a directory (preferably "AGENT").

4. Unpack the zip file to this directory.

```
$ unzip openvms-<CPU>.zip
```

The `[AGENT]` directory is called the *InstallDir* and the [BIN] directory is called the *ExecutableDir*.

Creating Symbols

We recommend that you create symbols for the jtool programs so that they can be executed from within jobs, for instance in a procedure that is set up in `SYLOGIN.COM`. We recommend you add the following:

```
$ !
$ ! Example, replace SYS$USER with your own logical or path to redwood directory.
$ !
$ !
$ JTOOL=="$SYS$USER:[REDWOOD.AGENT.<version>.BIN]JTOOL.EXE TOOL"
$ JLINK=="$SYS$USER:[REDWOOD.AGENT.<version>.BIN]JTOOL.EXE LINK"
$ JSLEEP=="$SYS$USER:[REDWOOD.AGENT.<version>.BIN]JTOOL.EXE SLEEP"
$ JREGISTER=="$SYS$USER:[REDWOOD.AGENT.<version>.BIN]JTOOL.EXE REGISTER"
$ JEVENT=="$SYS$USER:[REDWOOD.AGENT.<version>.BIN]JTOOL.EXE EVENT"
$ JMESSAGE=="$SYS$USER:[REDWOOD.AGENT.<version>.BIN]JTOOL.EXE MESSAGE"
$ JDESCRIPTION=="$SYS$USER:[REDWOOD.AGENT.<version>.BIN]JTOOL.EXE DESCRIPTION"
$ JSECRET=="$SYS$USER:[REDWOOD.AGENT.<version>.BIN]JTOOL.EXE SECRET"
$ !
$ ! In case you need to pass arguments, e.g. non-default instance
$ !
$ NETWORKPROCESSOR=="$SYS$USER:[REDWOOD.AGENT.<version>.BIN]NETWORK-PROCESSOR.EXE"
```

Further platform independent configuration instructions can be found in the Configuring Platform Agents section. In particular, if you want to guard the platform-agent against intruders and use by different server instances we suggest that you create a *SharedSecret* using the instructions found there.

Starting the platform agent interactively

For debugging and initial testing purposes it can be useful to start the network-processor interactively. For instance, to enable debugging until the process server is configured, run the following whilst logged in as *redwood*:

```
$ DEFINE RW_LOGLEVEL DEBUG
$ RUN [.AGENT.<version>.BIN]NETWORK-PROCESSOR
```

Starting the platform agent at startup

Starting the platform agent at system startup is recommended. For OpenVMS, we suggest starting with a single instance ('Default') running on port 1555. To do this, simply create a `STARTUP_DEFAULT.COM` script that you call from `SYSTARTUP_VMS.COM`. Suggested content is:

```
$ RUN/UIC=[REDWOOD] SYS$USER:[REDWOOD.AGENT.<version>.BIN]NETWORK-PROCESSOR.EXE /OUTPUT=.... /PROCESS_NAME=NW
```

Configuring HP OpenVMS Process Servers

Use

The following tabs are used to configure a process server connected to an HP OpenVMS system in SAP CPS.

Process Server Tab

Name	**Value**
Name	Alphanumeric and underscores
Default Job Time Zone	Olson time zone. Type any letter combination in the name to restrict the list, for example `lin` for *Europe/Berlin*

Job Definition Types Tab

Job definition types are required to execute job definitions of a corresponding type.

Field	**Value**
Job Definition Type:	*DCL*
Job Definition Type:	*JobChain*

Parameters Tab

The repository needs to know the host name (or IP address) and port number of the platform agent. The default port is 1555 and should be kept that way, unless another installed program is already using the default port.

Name	**Value**
RemoteHostName	<host name or IP address of system running platform agent>

All other parameters are optional, you may specify them, but they have usable defaults. See process server parameters for a list of parameters. If you do not know the value of the *SharedSecret*, see the Configuring Remote Agents section.

Queue Providers Tab

The queues the process server will serve, that is execute jobs from, are listed here.

Field	**Value**
Queue	*System* (or your own defined queue)

Resources Tab

Resources link job definitions to two or more process servers to spread the load. Process servers which share a same resource are the only ones that can run job definitions which have that resource. You may choose any name for a resource. Jobs that require a specific resource are spread across all process servers that have that resource and serve the queue the job was submitted in.

Services Tab

Services are used to execute jobs. When you specify a job definition type, the corresponding service is added automatically. You only need to fill in this tab manually if you want to create a file event only process server.

IBM z/OS Process Servers

Use

Jobs can be scheduled on IBM z/OS by SAP CPS using two different methods.

The first method utilizes the IBM Communication Server FTP server interface to JES2. This has the advantage that no agent needs to be installed or maintained on the mainframe.

It does require a continuous *poll* for job status over the network, which introduces a bit of overhead. For this reason this method is only recommended for short running interface jobs, not running many simultaneously long running jobs.

The second method installs a native platform agent on the host system. This requires JES2 and SMF user exits to get feedback on the submitted jobs. This has the advantage that the overhead introduced is lower so it is scalable to situations where job frequency is high, but it is requires installation of the software on the mainframe.

Two separate license keys govern which method(s) you are allowed to use:

License key	Description
ProcessServerService.JCL_FTP=true	Allows JES jobs to be run via ftp using a Windows or Unix platform agent.
ProcessServerService.JCL.limit=nn	Allows JES jobs to be run directly on the mainframe; SAP CPS software needs to be installed on the mainframe.

Job execution using CS FTP server

The CS FTP server method uses the extensive support for JES2 built into the CS FTP server. It is an *agentless* form of job execution. These jobs are *proxied* on a UNIX or Microsoft Windows platform agent that runs the JCL_FTP job runner. The job runner connects to the host via FTP, switches the FTP connection to JES mode, and then submits the job and waits for the job to finish.

See Using the JCL_FTP Job Definition Type for more information.

Limitations

The JCL_FTP method of running JES jobs currently has the following limitations when compared to the UNIX and Microsoft Windows platform agents:

- No support for file events.
- No support for monitoring system statistics (CPU, memory, I/O).

Job execution using native Platform Agent

The native z/OS platform agent runs jobs that have the JCL job definition type.

The SAP CPS platform agent submits JCL jobs to JES. The JES subsystem then runs the job. The platform agent also contains SMF and JES user exits that inform the scheduler server on the progress of the job. If the JES runs the job on a different node in the sysplex cluster than it was submitted on, the platform agent on the execution node will report the results of the job back to the scheduler server.

The z/OS platform agent consists of 5 components:

- Network Processor (JCSNWP)
- Central Master Area (CMA)
- Common Interface Area (CIA)
- SMF and JES Exits
- Utilities
- System Tools

The Network Processor (JCSNWP) is the equivalent of the network-processor part of the platform agent on UNIX and Microsoft Windows. It functions as the connector between z/OS and the &product Server. The JCSNWP runs as a started task (STC) under USS and starts a number of posix threads.

- One thread takes job information out of the CIA (which has a limited size) as quickly as possible and stores the data in USS file storage.
- Other threads form a TCP/IP server that is contacted by the scheduler server, and fulfills its request.
- If the server asks the agent to submit a job the job source is prepared: variables are substituted, nested JCL includes are read. The resulting JCL is sent to JES2 over the internal reader.
- If the server asks the agent whether there are any messages for it, the thread checks whether there are any persisted CIA occurrences and sends these to the server. If there are no messages the thread halts execution and waits for a signal from the CIA communication thread.

The SMF and JES Exits populate a common area named Common Interface Area (CIA) that is located in z/OS virtual storage (CSA) with information (occurrences) about the status of the z/OS system or rather the status of z/OS jobs and sequential files. The Central Master Area (CMA) is used to be able to locate the CIA.

The Utilities maintain the CIA, recreate occurrences that were lost and list the contents of all common storage (CMA-, CIA- and exit-storage).

The System Tools are ported programs that can be used to access the SAP CPS` API. This comes in the form of load modules that can be called from z/OS or USS. Identical tools exist on UNIX, Microsoft Windows and OpenVMS.

Installing the platform agent

The IBM z/OS specific installation instructions are found in the Installing the platform agent on IBM z/OS section.

Platform independent configuration instructions can be found in the Configuring Platform Agents section.

Creating the Process Server

Once the platform agent JCSNWP is running you can go to the scheduler UI and configure a process server that will contact the platform agent.

Information on the required values for the creation of IBM z/OS process servers can be found in the Configuring IBM z/OS Process Servers with the JCL Job Definition Type section of the documentation.

Limitations

The z/OS native agent currently has the following limitations when compared to the UNIX and Microsoft Windows platform agents:

- Limited support for file events (since M31). File events can only be raised when a *sequential* dataset (PS) is closed. Datasets that are libraries (PDS/PDSE), library members or files such as VSAM datasets cannot be triggered. Triggering events when such a file is complete can be done via the jevent system tool.
- Reading spool lists (job output) is only available on z/OS 1.10 and higher.
- No support for monitoring system statistics (CPU, memory, I/O).
- No support for auto-update.

See Also

- Using the JCL_FTP Job Definition Type
- Using the JCL Job Definition Type

Configuring IBM z/OS Process Servers with the JCL Job Definition Type

Use

The following tabs are used to configure a process server connected to an IBM z/OS system in SAP CPS that has a native platform agent installed. For jobs that run using the JCL_FTP Job Definition Type you do not need an additional Process Server; these run on a UNIX or Microsoft Windows process server.

Process Server Tab

Name	Value
Name	Alphanumeric and underscores
Default Job Time Zone	Olson time zone. Type any letter combination in the name to restrict the list, for example `lin` for *Europe/Berlin*

Job Definition Types Tab

Job definition types are required to execute job definitions of a corresponding type.

Field	Value
Job Definition Type:	*JCL*
Job Definition Type:	*JobChain*

Parameters Tab

The repository needs to know the host name (or IP address) and port number of the platform agent. The default port is 1555 and should be kept that way, unless another installed program is already using the default port.

Name	**Value**
RemoteHostName	<host name or IP address of system running platform agent>

All other parameters are optional, you may specify them, but they have usable defaults. See process server parameters for a list of parameters. If you do not know the value of the *SharedSecret*, see the Configuring Remote Agents section.

Queue Providers Tab

The queues the process server will serve, that is execute jobs from, are listed here.

Field	**Value**
Queue	*System* (or your own defined queue)

Resources Tab

Resources link job definitions to two or more process servers to spread the load. Process servers which share a same resource are the only ones that can run job definitions which have that resource. You may choose any name for a resource. Jobs that require a specific resource are spread across all process servers that have that resource and serve the queue the job was submitted in.

Services Tab

Services are used to execute jobs. When you specify a job definition type, the corresponding service is added automatically. You only need to fill in this tab manually if you want to create a file event only process server.

Configuring Platform Agents

Use

Platform process servers are made up of two independent but interacting parts:

- a process server in the Central Scheduling System
- a platform agent on the remote system

Both parts are started independently. The process server is usually started when the Central Scheduler System is started, or otherwise by a Start command issued via an API or the GUI. The platform agent is usually started as part of the operating system boot sequence.

Although either can run alone, the system is only functional when both parts are running and the platform agent service of the process server has established a TCP connection to the platform agent. When the connection gets interrupted, the process server will attempt to reconnect to the platform agent.

A connection must be established for jobs to run, file events to fire and monitoring data to be refreshed.

Most settings for the platform agent are stored in the repository as process server parameters. Settings that control who can do what are set on the remote system, as they need to be under the control of the individual system administrator of the remote system that the platform agent is running on.

Refer to the Process Server Parameters section for more information on process server parameters that are stored in the repository.

Configuration

On Microsoft Windows systems, the *Scheduler Service Manager* allows you to configure some options of the platform agent from within a user interface; see the Configuring Platform Agents on Windows section for more information.

Every platform agent instance created using the *Scheduler Service Manager* consists of a Windows Service that directly executes the `network-processor.exe` program. The service name will be *Scheduler Platform Agent <instance> <version>* and the technical name will be `jcs_<version>_<instance>`.

The configuration files are stored in `${InstallDir}/net/`. The `net` directory can contain subdirectories so that multiple platform agents can be managed from a single tree. The directories are searched in the following order:

1. `net/instance/<instance>/<file>`
2. `net/hostname/<hostname>/<file>`
3. `net/global/<file>`

In other words, instance specific settings go before hostname specific settings, and hostname specific settings go before global settings.

In the above locations, the following variables were used:

- <instance> is the name of the instance, which by default is set to `default`.
- <hostname> is the hostname of the server, as returned by the command `hostname`.
- <file> is the name of the file that it is looking for.

The files that the system looks for are detailed below.

File	Use
`port`	The port the agent listens on for inbound connections
`server_acl`	The Central Scheduler the platform agent is locked to
`address_acl`	The hostname(s) or IP addresses of the Central Scheduler the platform agent is locked to
`listen`	The IP addresses that the platform agent should listen on
`private/secret`	The secret for authentication
`chown`	A symbolic link to the chown binary, improving security when using sudo User Switching Security Mode

Note

You must install the SAP CPS platform agent on a local file system; SAN file systems might be considered local, when they are mounted as iSCSI, for example.

port

The port the platform agent will use at startup is saved in a file named `port`. If no such file is found, the default of `1555` will be used.

The only contents of the `port` file should be the port number; to set the port number for instance `production` to `1566`, you can proceed as follows:

On Unix

```
echo 1566 > /opt/redwood/net/instance/production/port
```

Note that `/opt/redwood` is the installation directory.

On Windows

```
echo 1566 > G:\redwood\net\instance\production\port
```

Note that `G:\redwood` is the installation directory.

server_acl

The `server_acl` file, if it exists, limits the platform agent to only talking with Central Schedulers that have a system ID that is on the list in the server_acl file. To find out what a system's SystemID is, log in to the system and observe the browser heading; the part before the [character is the system ID. Any characters in the system ID that are not alphanumerical, such as '-' dashes, should be converted to underscores: _

address_acl

If set, the `address_acl` file will limit which IP addresses can connect to the server. The file can contain a list of IP addresses, hostnames and/or IP ranges.

Example:

```
#
# Example address_acl file
#
192.168.10.0/24
10.31.0.0/255.255.0.0
cps1.prod.sap.de
cps2.prod.sap.de
```

listen

You use the `listen` parameter to specify which IP addresses of the platform agent's computer are used to accept new connections. By default it is "0.0.0.0" and accepts any connection. You can limit this to a particular IP address or hostname, which resolves to a local IP address.

private/secret

If set, it should contain a secret that the Central Server also has configured for this process server. The secret is used to create a hash function over the content of the message being passed. If both sides do not possess the same secret, the agent log file will contain messages such as these:

```
error <date> [***-http-request #** tid=***] http.http - Content digest *** does not match computed value ***
error <date> [***-http-request #** tid=***] http.http - Request with content has incorrect HMAC checksum
```

To correct this, make sure both sides possess the same shared secret.

When you install the platform agent using the installer, the shared secret is generated during the installation. When you registered the platform agent during the installation the shared secret will also be known to the Central Scheduler. If not, you have to paste the value into the *SharedSecret* process server parameter and restart the process server.

You can generate the shared secret with the jsecret executable.

On Microsoft Windows you can also use the *Scheduler Service Manager* to set the secret; see the Configuring Platform Agents on Windows section for more information.

Improving Security when using sudo

The SAP CPS installer creates a sudo configuration for the SAP CPS user when you choose *sudo* as your user-switching mode. This could be used by a user to gain access to files owned by `root`. To avoid this, SAP CPS allows you to specify your own `chown` command. SAP CPS ships with an example `chown.sh` which checks parameters.

The `chown` file in the net directory is a symbolic link to the `chown` binary as detected by the installation routine. You can create a symbolic link to the `chown.sh` script in the SAP CPS `bin` directory to improve security. Edit the `chown.sh` script to suit your security needs.

Logging

The agent logs to a file. See Agent Logging for more information.

Unix Parameters

UNIX specific parameters consist of two parts: those used by the `platform-agent` script and those used by the `network-processor`.

Parameters for the platform-agent script

File	**Use**
{{${InstallDir}/etc/startup/${instance}/user}}	the UNIX account that the script switches to.
{{${InstallDir}/etc/startup/${instance}/net}}	Overrides the net directory location (default: `${InstallDir}/net`).
{{${InstallDir}/etc/startup/${instance}/version}}	Which version is used for this instance.
{{${InstallDir}/etc/startup/${instance}/loglevel}}	What loglevel the platform-agent and initial network-processor use.

user

Normally the platform-agent and network-processor processes do not run as root to prevent security issues. During installation a choice was made as to which user switching mode is desired and under which account the platform-agent and network-processes will run during their lifetime. The account that the platform-agent shell script switches to when it is called by the `.../etc/scheduler` startup script is stored in `${InstallDir}/etc/startup/<instance>/user`.

net

The *net* directory can be changed. This is not supported, and should not be used. This is documented for completeness purposes.

version

The platform-agent supports multiple instances and multiple versions from the same `${InstallDir}` which allows auto-upgrade and auto-downgrade of the platform agent whenever the Central Scheduler changes its version. When you install the agent, the version that is started is stored in {{${InstallDir}/etc/startup/${Instance}/version}}.

loglevel

The loglevel setting is explained in Agent Logging. The logfile used by the platform-agent script is fixed to `${InstallDir}/etc/log/${Instance}/${Instance}.log`.

Parameters for the network-processor executable

The UNIX specific parameters for the network-processor}} executable are kept in the {{ ${InstallDir}/net }} hierarchy, just like the system independent settings.

Some items reside in a further `private/` subdirectory. For security reasons these should only be readable by the user that the `network-processor` runs as.

File	Use
`private/whitelist`	list of users that jobs can be run as
`private/blacklist`	users that cannot be used for running jobs
`usermode`	mode used to switch accounts

private/whitelist and private/blacklist

On Unix, it is common practice to prevent certain users from being able to log on interactively. You can also avoid jobs to run as specific users. Do this by providing SAP CPS with a list of authorized or banned users. These settings are saved in the {{ ${InstallDir}/net }} hierarchy, in the `private` sub-folder. For security reasons they should only be readable by user `redwood` and `root`.

If you provide a whitelist then the blacklist is not used. The default value is a blacklist containing `root,daemon,bin,sys,adm,uucp,nuucp,lp,listen,sysadm,smtp,ftp,tftp,news,sysdiag,sundiag`.

usermode

The user-switching mode that the network-processor uses to run jobs under the correct account is stored in `${InstallDir}/net/.../usermode`. It contains one of the following: `plain`, `root`, `sudo` or `setuid`.

Configuring Platform Agents on Windows

Use

The installer will install and start a service for the platform agent. You can manage this service using one of the following:

- *Microsoft Windows Service Manager* - the services are named *Scheduler Platform Agent <instance> 0_28_49*. Access the Service Manager via *Start* > *Settings* > *Control Panel* > *Administrative Tools* > *Services*. The string 0_28_49 may differ in your installation as it reflects the internal version number of the platform agent.

- *Scheduler Service Manager* - also allows you to install and uninstall agents as well as start, stop, and configure existing agents. You can find the *Scheduler Service Manager* at *[All] Programs* > *Scheduler* > *Scheduler Service Manager* in the start menu.

Note

The installer will create the Windows service and have it run under the `NT Authority\System` account. This is required to have the platform agent update automatically. If you want to run the platform agent service as another user, you can use a regular user with the *SeAssignPrimaryTokenPrivilege* and *SeIncreaseQuotaPrivilege* privileges. SAP recommends to use the automatic update feature for platform agents.

The following languages are available in the *Scheduler Service Manager*:

- *English*
- *German*

The *Scheduler Service Manager* allows you to install, start, stop, configure and uninstall platform agents on the local system. When you start it, it lists the platform agents and displays the *Status*, *Version*, as well as the *Port* used by the platform agents.

Installation

For instructions on how to install a Microsoft Windows platform agent, please consult the Creating a Microsoft Windows Process Server section of the documentation.

Configuration

The *Scheduler Service Manager* allows you to add, remove, stop, start and configure platform agents.

- *Start*/*Stop* - Allows you to start or stop a service
- *Uninstall* - Allows you to delete the selected platform agent service
- *Refresh* - Refreshes the status of all services
- *Install* - Allows you to install a new platform agent
  - *Preferred instance name* - Allows you to specify the instance name
  - *Port* - Allows you to specify a port; ports must be unique, no two platform agent services can run concurrently listening on the same port
  - *Enable auto-update* - Allows you to specify if the platform agent you be updated as soon as it connects to a central server of a higher version
  - *Account* - Allows you to specify an account to be used by the service
- *Exit* - Quits the program, the platform agent services are not affected

Each service has a context menu which allows you to configure the selected platform agent service.

- *Start*/*Stop* - Allows you to start or stop a service
- *Configuration* - Allows you to configure the platform agent
  - *Port* - Allows you to change the port used by the platform agent
  - *Enable auto-update* - Allows you to specify if the platform agent you be updated as soon as it connects to a central server of a higher version

    - *Log level* - Allows you to set the log level of the platform agent
    - *Logfile* - Allows you to specify a log file. Note that the *Scheduler Service Manager* cannot validate log file directories which contain substitution parameters. This means that when the path does not exist the *Scheduler Service Manager* will not warn you.
    - *Generate secret* - Allows you to generate a new secret for the platform agent.
    - *Restart service* - The platform agent service will be restarted, by default.
- *Choose language* - Allows you to switch between German and English.
- *Get support assist file* - Generate an XML file which can be useful for troubleshooting issues.

The configuration dialog is available in the context-menu of the platform agent and allows you to change the port, set the log level and log file as well as generate a new secret. If you change the port or secret you should also update the process server in the Central Scheduler server.

Please see Configuring Platform Agents for more information on the different options for log levels and log files.

Note that the *Scheduler Service Manager* cannot validate log file directories which contain substitution parameters. This means that when the path does not exist the *Scheduler Service Manager* will not warn you.

Procedure

Setting a log level on the platform agent

1. Choose to *Scheduler* > *Scheduler Service Manager* in the Windows start menu.
2. Choose *Configure* from the context menu of the platform agent you would like to debug.
3. Enter a log level into the *Log Level* field.
4. Enter the full path of the output file into the *Logfile* field, make sure *Restart service* is checked, and choose *OK*.

Result

The platform agent will be restarted and the log file will be filled with output corresponding to the log level.

Example

Setting a log level of Warn on the `opsys` group of the `network-processor` and `job-processor`

1. Choose *Scheduler* > *Scheduler Service Manager* in the Windows start menu.
2. Choose *Configure* from the context menu of the platform agent you would like to debug.
3. Enter `network-processor.opsys=Warn;job-processor.opsys=Warn` into the *Log Level* field.
4. Enter `C:\debug_${ProcessId}.log` into the *Logfile* field, make sure *Restart service* is checked, and choose *OK*.

Configuring Platform Agents on Unix

Use

The installer will install a set of executables for the platform agent. You use the `rca` script to maintain instances of platform agents in an installation directory.

The following languages are available in the `rca` script:

- *English*
- *Deutsch*
- *Nederlands*
- "Français

The `rca` script allows you to install, configure and uninstall platform agents on the local system.

Installation

For instructions on how to install a Unix platform agent, please consult the Creating a Unix Process Server section of the documentation.

Configuration

The configuration option is available once you have selected an instance (option `2`) and allows you to change the port, set the log level and log file as well as generate a new secret. If you change the port or secret you should also update the process server in the Central Scheduler server.

Please see Configuring Platform Agents for more information on the different options for log levels and log files.

Procedure

Creating an instance

1. Run `rca` in `<install_dir>/0_28_49/bin/` as user `root`.
2. Enter `1` for *English*.
3. Enter `3` for *New Instance* (note that if there is already an instance selected, the number will to enter will be `4`)
4. Enter an instance name.
5. Enter `3` for *Configuration*, change the appropriate settings like *Listening port*, *Account to run as* and optionally create a shared secret with `8` *Create Shared Secret*.
6. Enter `10` to exit from the script.
7. Run `sudo /etc/init.d/scheduler restart` to restart the network agents.

Setting a log level on the platform agent

1. Run `rca` in `<install_dir>/0_28_49/bin/` as user `root`.
2. Enter `1` for *English*.

3. Enter `2` for *Instance Name* and enter the number for the relevant instance. Note that the current instance is now printed next to *Instance Name*.
4. Enter `3` for *Configuration*, enter `6` for *Change Loglevel*.
5. Enter a number for the desired log level.
6. Enter `10` to exit from the script.
7. Run `sudo /etc/init.d/scheduler restart` to restart the network agents.

Note

When you run the `rca` script as another user than user `root`, your functionality is limited.

Result

Creating an instance

An instance with the desired name and configuration has been created and started.

Setting a log level on the platform agent

The platform agent will be restarted and the log file will be filled with output corresponding to the log level.

Example

Setting a log level of *warn*

1. Run `rca` in `<install_dir>/0_28_49/bin/` as user `root`.
2. Enter `1` for *English*.
3. Enter `2` for *Instance Name* and enter the number for the relevant instance. Note that the current instance is now printed next to *Instance Name*.
4. Enter `3` for *Configuration*, enter `6` for *Change Loglevel*.
5. Enter `4` for the *warn* log level.
6. Enter `10` to exit from the script.
7. Run `sudo /etc/init.d/scheduler restart` to restart the network agents.

Configuring Load Balancing on Platform Agents

Use

When your jobs that can run on more than one process server, you will want to spread the load as evenly as possible. By default the system will run a new job on the server that has the lowest amount of running jobs. In other words, if three process servers are eligible and they are running 1, 2 and 3 jobs respectively the system will attempt to run the job on the agent with 1 running job.

You can optionally also use *OS metric load balancing*, where the decision is no longer based on the number of running jobs but on some *metric*. The metric that is chosen is user-defined. All metrics are first stored in the *monitor* data. This is a tree of nodes and values.

Built-in platform agent metrics

The platform agents are able to send *CPU* and *memory* usage data to the server. The frequency at which this data is sent is controlled with the *MonitorInterval* process server parameter. If you do not set this parameter, the default value is 60 (seconds). Set the value to 0 (zero) if you do not plan to use load-balancing or use different values than the system metrics.

Once you set MonitorInterval to a non-zero value the data can be used for load balancing:

```
/System/ProcessServer/<process_server>/Performance/CPUBusy
/System/ProcessServer/<process_server>/Performance/PageRate
```

Custom defined metrics

The load balancing system can also use other metrics. These can be written to the monitoring tree using the *Java API*, or using the jmonitor tool within a (possibly long-running) job.

For example, if you want to base your decision on the number of free printers available at a particular system, you should call the following in your job(s) whenever the number of available printers changes:

```
jmonitor -j /System/ProcessServer/<process_server>/Performance/FreePrinters=$FREE
```

Setting the load factor

You influence the computed load on a particular process server by adding one or more *load factors* to its definition. Each load factor has three attributes:

- Threshold - the maximum allowed value, once reached, the process server is put to *Overloaded* until the threshold is no longer met.
- Multiplier - the multiplier is used to compare the current value to other process servers.
- Monitor Value - the monitoring leaf value that is used.

Once you have set the load factors the system will choose a new job to run on the system that has the lowest value for the following equation:

```
sum(Multiplier * Monitor Value)
```

If a process server has at least one load factor where the current monitor value exceeds the threshold value it is not chosen.

Note

If any of the eligible process servers has no load factors at all then OS metric load balancing is not used and the system reverts to counting the number of already executing jobs.

Procedure

Setting up load balancing between two agents

1. Navigate to *Environment* > *Process Servers*
2. Choose *Edit* from the context menu of the weak process server A.

3. On the *Load Factors* tab choose *New* and enter *System load* into the description field.
4. In the *Multiplier* field enter `2`.
5. In the *Threshold* field enter `90`.
6. In the *Monitor Value* field choose the *CPUBusy* monitor for the path */System/ProcessServer/<agent>/Performance* where <agent> is the agent that you are editing.
7. Choose *Save & Close.*
8. Choose *Edit* from the context menu of the strong process server B.
9. On the *Load Factors* tab choose *New* and enter *System load* into the description field.
10. In the *Multiplier* field enter `1`.
11. In the *Threshold* field enter `75`.
12. In the *Monitor Value* field choose the *CPUBusy* monitor for the path */System/ProcessServer/<agent>/Performance* where <agent> is the agent that you are editing.
13. Choose *Save & Close.*

Example

You want to balance the workload between two platform agents running on two different machines. The first machine has two slow CPUs, and the other one has 4 fast CPUs that are each twice as fast. You want to maximize throughput of the system. You can do this by using the bigger server to good effect by allocating more jobs to it, but also need to reserve capacity on that machine for on-line users. To do so you could implement the following settings:

Process server A (slow, with 2 CPUs) suggested load factor:

Field	**Value**
Multiplier	1
Threshold	100
Monitor Value	CPUBusy (for that process server)

Process server B (fast, with 4 CPUs) suggested load factor:

Field	**Value**
Multiplier	1
Threshold	75
Monitor Value	CPUBusy (for that process server)

The multiplier value can remain 1, as a job running on the faster system B will have less of an impact on CPU load than the same job will have on system A. Multipliers are most often used when you combine multiple load factors.

The threshold is set to 75 on process server B, so that we reserve one CPU (one out of four = 25%) for non-batch work.

Automatically Updating Platform Agents

Use

When the Central Server is updated to a new version the platform agents need to be upgraded as well. The software tries to make this as painless as possible. At the moment the UNIX and Microsoft Windows platform agents can update themselves, if certain conditions are met. This functionality is not available for the HP OpenVMS and IBM z/OS agents.

The Central Server updates the database and other parts of the central system the first time that you run the new version. After this is complete it will (try to) start the platform agents that are marked as having automatic start. As part of the startup procedure the server sends a *configuration* request to the platform agent, telling it which process server parameters it is supposed to use. As part of this command it also sends its own version ID. The platform agent compares this version ID with its own version ID. If the two do not match it sends a result saying that the configuration failed because the version did not match. It will also tell the server whether it does have the required version installed (but apparently not active for this instance) and whether it is able to restart other versions.

If the server finds that the agent is able to restart but does not have the same version installed it then sends the installer for the platform agent to the existing platform agent. The platform agent stores the installer on the file system and runs the installer in non-interactive mode. The installer installs the software for the agent side-by-side to the existing installation.

Once the software is successfully installed, or if the agent mentioned that the software was already installed but not active it sends a *restart* request to the agent. The agent activates the new software version and then, as they say, 'magic happens'. The magic is that the old agent stops and the new agent starts. How this is accomplished differs on the platform, and is detailed below.

Requirements

For the auto-update to function, you must install the agent in the proper manner and start it in the proper manner.

For Microsoft Windows, make sure that the agent runs as a service running as the `Local System` account. Only this account is able to run the installer. This is easy to do during the first time install of the platform agent: just keep the *Auto update* check box selected during the installation.

For UNIX systems the most important requirement is that the platform agent is started by means of the `${InstallDir}/<version>/bin/platform-agent` shell script. If this is done then the system can update itself.

The administrator can control whether auto-update is performed by creating or removing the `${InstallDir}/etc/startup/${instance}/autoupdate` directory. This is where the auto-update process will store the downloaded installer for the missing version. If this directory does not exist, the auto-update procedure is disabled. Normally, this directory is created by the installer when you answer positively to the question whether to allow auto-update. If you did not, and do not wish to re-run the installer you can enable this by creating this directory manually. Make sure the privileges are such that the network-processor program can write to that directory.

Availability

Available as of M25. Improved in M26, M26.7 and M27.^M

How it works - UNIX

When a platform agent is configured as version X, and version Y is requested by the server, the network processor first writes the correct version to file `${InstallDir}/etc/startup/${instance}/version`. Then it stops using a magic exit code. The `platform-agent` script recognises this and immediately restarts the `platform-agent` of the correct version.

For this to function you should have started the agent via the `platform-agent` script directly or via the boot scheduler script (recommended).

How it works - Microsoft Windows

On Microsoft Windows the agent needs to stop its own service and start the service of the new agent. It does this by calling the *Scheduler Service Manager* and then waiting for a stop command. The service manager, when it succeeds in creating the new service, will stop the old service and then start the new service. If the service manager fails, it does not send the stop command to the old service, which will just keep running.

Tuning

New in M27.

When the installation has a larger number of platform agents the system will serialize the process of sending the updated platform agent installer binaries, as these are quite large. The number of simultaneous updates is governed by the registry parameter `/configuration/PlatformAgent/SimultaneousUpdates`. Change this as required, but make sure the value is at least 1 (one). The default value is 5.

Securing Communications for Platform Agents and System Tools

Use

Redwood supplies an optional communications library that accesses an internal API within the platform agent code to intercept all network communication. If this library is installed it becomes possible to mandate secure encrypted communication with the Java server.

Whether you can obtain this library is subject to import laws of your country.

If you are unable to install this library because of legal restrictions within your jurisdiction or company, the product is not unsafe -- all connections are still authenticated; the agent will not talk to everyone, only those parties that know the communication secret can do so.

You do have to safeguard access to the network traffic though as potentially valuable information is sent across the network.

The optional communications library that Redwood provides is based on OpenSSL, which is the same technology that all major UNIX operating system suppliers deliver for such encryption. To facilitate easy deployment we supply a compiled version that plugs the OpenSSL security code into the network connections that the network agent and the JFTP tool make. You do not need to worry about matching library versions on different systems etc.

Obtaining the library

The optional library can only be obtained under a direct contract with Redwood Software. Contact Redwood [http://www.redwood.com/] for more information. Redwood will supply you with an additional license and the library.

To install the library put the file `secure.so` or `secure.dll` that is appropriate for your platform in the agent's `bin` directory.

Using secure JFTP

As soon as you have put the shared library in the appropriate directory jftp will add new options such as `-tls` and `-ssl`. Type an incomplete jftp command to see the additional syntax. To get a list of all allowed ciphers use the following command:

```
jtool ftp -tls -cipherlist ALL
```

Since this contains no actual command the tool will spell out all options, including the enabled ciphers. Since you enabled all ciphers, this gives a list of all possible ciphers.

Configuring the Platform Agent

If you want the platform agent to act as a HTTPS server instead of a HTTP server you have to add configuration files. You can put these in the `net/instance/<instance>`, `net/hostname/<hostname>` or `net/global` directory, depending on which instances they should apply to. The platform agent only supports TLS v1.0, not SSL v2 or v3 as the only HTTP client that it is supposed to talk to is the Central Server, and that is able to speak TLS v1.0 as well.

Platform Agent with Anonymous Diffie Hellman cipher

An anonymous Diffie Hellman cipher will provide encryption. Authentication is done via the standard SAP CPS mechanism of shared keys. The difficulty of cracking the protocol depends on the chosen cipher.

- To make the platform agent use the TLS protocol, add a file `extralib`. Put the single word `secure` in this file.
- Put Diffie Hellman data in a `dh1024.pem` file.
- Decide on the ADH ciphersuites that you want to enable, and put this in the file `cipherlist`. The simplest form is to put the word `ALL` in this file. Use the `jftp -tls -cipherlist ALL` command to see which ciphers are available.

The `dh1024.pem` file can be generated using the following openssl command (not supplied):

```
openssl dhparam -out dh1024.pem 1024
```

Platform Agent with RSA or DSA cipher

An RSA or DSA key and certificate can also be used. They provide authentication of the client and/or the server to the other party, but only if this is explicitly enabled. If you do not enable these, there is no security advantage over Anonymous Diffie Hellman.

- To make it use the TLS protocol, add a file `extralib`. Put the single word `secure` in this file.
- (For DSA only:) put Diffie Hellman data in the `dh1024.pem` file.
- Put a public server certificate in the file `rwscert.pem`.
- Put a private server key in the file `private/rwskey.pem`.
- If the private server key requires a passphrase, put the passphrase in the file `private/passphrase`.
- Decide on the ciphersuites that you want to enable, and put this in the file `cipherlist` if you want to use a different cipherlist than the default.

To enable the advantages of a RSA/DSA key over ADH, configure the server's CheckAgentHostname and/or CheckAgentCertificate process server parameters, as discussed below.

For testing purposes you can use a self-signed SSL certificate. Run the following openssl command (not supplied) to generate these.

```
cd ..../net/instance/default
openssl req -new -x509 -nodes -out rwscert.pem -keyout private/rwskey.pem
```

A signed SSL certificate can be generated as follows:

- Create a RSA private key for your Apache server (will be Triple-DES encrypted and PEM formatted).

```
cd ..../net/instance/default
openssl genrsa -des3 -out private/rwskey.pem 1024
```

- Copy the pass-phrase you entered in the above command to `private/passphrase`.
- Create a Certificate Signing Request (CSR) with the server RSA private key (output will be PEM formatted). Make sure you enter the FQDN ("Fully Qualified Domain Name") of the server when OpenSSL prompts you for the "CommonName", for example when you generate a CSR for an agent which will be later accessed via RemoteHostName `www.foo.dom`, enter "www.foo.dom" here.

```
openssl req -new -key private/rwskey.pem -out request.csr
```

- You now have to send this Certificate Signing Request (CSR) to a Certifying Authority (CA) to be signed. Once the CSR has been signed, you will have a real Certificate. You can have a CSR signed by a commercial CA, or you can create your own CA to sign it.
- Copy the `request.crt` file to `rwscert.pem` in the appropriate directory.
- The `request.csr` file is no longer needed.

Note

All security data files are in PEM format.

Configuring the Central Server

Since SAP CPS runs on a Java 1.4.2 or higher JVM, it already possesses the required encryption code. When the process server talks to a platform agent that is configured to use TLS it will automatically upgrade to the HTTPS protocol. This provides encryption when it is set up in the agent. It does not guarantee encryption, and uses the standard scheduler mechanism of shared keys for authentication.

There are three process server parameters that enforce encryption by the server, and allow additional authentication.

Parameter	Use	Default
SecureConnection	If set to true the server will refuse to talk to an agent that is not secure. If set to false it will allow the platform agent to make this decision	false
CheckAgentHostname	Should the JVM check the hostname in the certificate presented by the platform agent? If set to true, the RemoteHostName used to connect to the platform agent must be the same FQDN as that presented in the agent's certificate	false
CheckAgentCertificate	Should the JVM check the check the certificate presented by the platform agent? If set to true, the JVM truststore must contain the agent's public certificate and the root CA certificate	false

License

The library is based on OpenSSL and is licensed according to a dual Apache style license. We reproduce the entire license in the Attributions section.

Command Line System Tools

Use

The Platform agents for UNIX, Microsoft Windows, OpenVMS and z/OS (See section below regarding z/OS) ship with the following system tools:

Tool	**Purpose**	**Connection file**	**Job-context**	**Available on z/OS**
jdescription	Set the description of the current job	-	Mandatory	-
jevent	Raise an event	Yes	-	Yes
jftp	Transfer files via FTP	-	-	-
jgetpar	Get configuration parameter data	-	Mandatory	-
jlink	Link files to jobs	-	Mandatory	-
jmessage	Send an operator message	-	Mandatory	-
jmonitor	Send monitoring data values	-	Mandatory	-
jregister	Register a process server	-	-	-
jsecret	Generate a user or network-processor secret	Yes	-	Yes
jscript	Run a script	Mandatory	-	Yes
jsleep	Sleep a number of seconds	-	-	Yes

All tools are available in two forms: the `j<tool>` form such as `jftp` and as a sub-option of `jtool` such as `jtool ftp`. Both are available after the installation. If you do have `jtool` but not a particular `j<tool>` form, re-run the `jtool install` command.

File Paths

You might want to use these utilities outside of SAP CPS jobs; in this case, you should take the following into account:

Since the path to the `bin` directory contains version information and that this will change when a new version is installed, SAP recommends using environment variables (all platforms) to contain the version that your scripts should use. You can also use symbolic links on UNIX and Microsoft Windows Vista or later, re-parse points/junctions on earlier versions of Windows (XP, 2003), and Logicals on HP OpenVMS.

Connection file

Some tools are suited to communicating with the Java server over the server's HTTP interface. This is useful when you want to run commands outside of the scheduler environment, for instance from a server where you have not installed the platform agent. It is explicitly allowed to copy the `jtool` binary over to a different machine and call it from there. Such tools require the use of a *connection file* that can be created using jsecret.

Job-context

Other tools are meant for use within a SAP CPS OS job executed by the platform agent. This is called running in *job-context*. Such tools use the `-j` or `--jobcontext` argument to verify that you understand that they must be called in job-context. At a later date the tools may be extended to also use a connection-file instead, so the job-context argument must be used for upwards compatibility with such a future possibility.

Available on z/OS

On z/OS most system tools are not available.

- As z/OS jobs run as JES jobs there is no *job-context*. System tools that depend on the job context are not supported.
- jftp has not been ported to function properly in an EBCDIC environment yet.
- jregister is mainly used by the installation script; since that does not exist on z/OS we have not ported jregister either.

Specific instructions on how to use the system tools on z/OS are documented in z/OS System Tools.

Logging

All tools, including the platform agent itself, log in a common logging format and use the same arguments to overrule the default logging settings. See Agent Logging Settings for more information.

jdescription, jtool description

Use

A platform agent tool used to update the job's description. You can call it as `jdescription` or `jtool description`

This tool cannot be used outside job-context at the moment, so the `-j` or `--jobcontext` parameter is mandatory. As with all job-context operations, the update is performed asynchronously so there is no guarantee that the operation will succeed in the server.

Syntax

```
jdescription {-j | --jobcontext} <description>
jtool description {-j | --jobcontext} <description>
```

Argument	**Description**
-j, --jobcontext	Run the command in job-context
<description>	New description for the job

Example

The following UNIX shell example shows how to set the job's description as progress of the job continues.

```
percentage=0

while [ ${percentage} -ne 100 ]
do
  jdescription -j "Starting operations... ${percentage}%"
  jsleep 5

  percentage=$((${percentage} + 10))
done
```

jevent, jtool event

Use

A platform agent tool used from a command line inside or outside a job to raise a named event inside SAP CPS.

Syntax

```
jevent [-s] <connection-file> <eventname> [<comment>]
```

Argument	Description
-s	Silent, no output will be generated
<connection file>	A file containing connection details, like host, port, username, password, isolation group that is created using jsecret
<event name>	The name of the event to raise
<comment>	The raiser comment

Example

In this example, the SAP CPS server is named `pr1.masalan.com`, its port is `50000` and the isolation group to use is `GLOBAL`. Please note that if you want to use the isolation group `GLOBAL`, you do not have to specify it as by default, the `GLOBAL` isolation group will be used. Note also, that when you enter your password for the connection file, no text is displayed as you type; this improves the security of your password.

```
D:\Agent\0_28_49\bin>jsecret -c ..\..\net\instance\default\masalan.conn pr1.masalan.com 50000 GLOBAL
 Username:jdoe
 Password:

D:\Agent\0_28_49\bin>jevent -s ..\..\net\instance\default\masalan.conn "FinancialsLoadedSuccessfully" "Event
Login successful on pr1.masalan.com
Successfully raised event with ID 74

D:\Agent\0_28_49\bin>
```

Note

SAP recommends you store your connection files in the directory for your instance.

jftp, jtool ftp

Use

This is the platform agent non-interactive ftp tool; all options are specified directly on the command line. The jftp tool will return error code 1 when an error occurs. Note that native UNIX and Windows ftp run interactively and do not return error code 1 when something unexpected happens. JFTP will also return a code that indicates the issue, for more information on return codes see the FTP return codes section of the documentation. The jftp tool will return standard FTP return codes when the FTP server does, so the Internet is a great resource to look-up the meaning of a specific error.

jftp adheres to the following RFCs:

- RFC 959 [http://www.ietf.org/rfc/rfc959.txt] - File Transfer Protocol.
- RFC 2228 [http://www.ietf.org/rfc/rfc2228.txt] - FTP Security Extensions.
- RFC 2246 [http://www.ietf.org/rfc/rfc2246.txt] - The TLS Protocol Version 1.0.

- RFC 4217 [http://www.ietf.org/rfc/rfc4217.txt] - Securing FTP with TLS.

If you have enabled secret file transfers, but do not have OpenSSL installed on the host where jftp runs, you will get an error that SSL support is not available. Please refer to Secure FTP for information on how to enable secure file transfer on your platform.

You no longer have to hard-code the username and password or provide credentials in your job definitions, you can use a "secret" file. Since credentials cannot be exported, a "secret" file is also useful when you have multiple SAP CPS environments. You generate secret files with jsecret, using the `-c` option. Note that for FTP, the default port is `21`.

Usage

```
jftp [-]command [ [-]command... ]

  command ::=
    get <options> <login arguments> <outfile> <infile> |
    put <options> <login arguments> <infile> <outfile> |
    list <options> <login arguments> <directory> |
    delete <options> <login arguments> <file> |
    mkdir <options> <login arguments> <directory> |
    rmdir <options> <login arguments> <directory> |
    quote <options> <login arguments> <directory>

  <options> ::=
     [-pasv][ -ascii | -binary ] | -use_cwd | -port <port_number>
      | -tls | -ssl | protect | -cert <file> | -key <file> | -l <logfile> | <options> ]

  <login arguments> :=
    { <host> <user> <password>| -secretfile <secretFile> }

Keywords and parameters
<infile>  The file on the local computer.
<outfile>  The file on the remote system.
<file>  The file on the remote system.
<directory>  The directory on the remote system.
<host>  The hostname of the FTP server.
<user>  The user name used to log on to the remote server.
<password>  The password to log on (if environment variable JFTP_PASSWORD is set this is optional).
tls  Use a TLS v1 encrypted control connection.
ssl  Use an SSL v3 encrypted control connection. Redwood recommends that you use tls.
protect  Use TLS/SSL for the data connection.
cert <file>  The location of the client certificate.
key <file>  The location of the client key.
secretfile <secretFile> The location of the secret file generated with "jsecret -c"
pasv  Use a passive FTP connection.
```

When you use TLS/SSL the control connection is always encrypted. When the data must also be encrypted, you must use the -protect option.

Note

<login arguments> can only be specified for the first command. <options> can be specified for all commands.

Note

You must use a numerical value if you set the port number using the -port option.

Examples

```
jftp get -ascii myhostname myusername mypassword myremotefile mylocalfile

jftp put -ascii myhostname myusername mypassword mylocalfile myremotefile

jftp quote myhostname myusername mypassword "site filetype=jes"

jftp quote myhostname myusername mypassword "site filetype=jes" put -ascii mylocalfile
         myremotefile get myremotefile mylocalfile.check

jftp get -port 21 myhostname myusername mypassword sourcefile destinationfile

jftp put -tls -key mykey.pem -binary host user password filein fileout

jftp put -protect -tls -key mykey.pem -binary host user password filein fileout

jftp put -secretfile ftp.sec filein fileout

jftp put -secretfile ftp.sec -protect -tls -key mykey.pem -binary filein fileout
```

JFTP Return Codes

Use

Return codes, also referred to as *reply codes*, give feedback on the status of the previously executed command. RFC's define the standards for protocols to allow tools to inter-operate, however, vendors are not forced to adhere to the RFC's standards and many don't adhere totally. The jftp tool will return standard FTP return codes when the FTP server does, so the Internet is a great resource to look-up the meaning of a specific error.

FTP reply codes are defined in RFC 354 [http://www.ietf.org/rfc/rfc354.txt] and RFC 414 [http://www.ietf.org/-rfc/rfc414.txt] and according to those RFC's, reply codes are made of the following:.

The first digit specifies the type of error:

- 4xx Unsuccessful reply. The request is correctly specified but the server is unsuccessful in correctly fulfilling it.
- 5xx Incorrect or illegal command. The command or its parameters were invalid or incomplete from a syntactic viewpoint, or the command its inconsistent with a previous command. The command in question has been completely ignored.

The second digit specifies the general category to which the response refers:

- x40 Secondary access. The primary server is commenting on its ability to access a secondary service.
- x5x FTP results.

The final digit specifies a particular message type. Since the code is designed for an automaton process to interpret, it is not necessary for every variation of a reply to have a unique number. This is why jftp does not return this digit.

The officially assigned reply codes above 400 and relating to FTP are:

- 400 This service not implemented.
- 401 This service not accepting users now, goodbye.
- 430 Log-on time or tries exceeded, goodbye.

- 431 Log-on unsuccessful. User and/or password invalid.
- 432 User not valid for this service.
- 433 Cannot store files without valid account. Enter account.
- 434 Log-out forced by operator action. Phone site.
- 435 Log-out forced by system problem.
- 436 Service shutting down, goodbye.
- 450 FTP: File not found.
- 451 FTP: File access denied to you.
- 452 FTP: File transfer incomplete, data connection closed.
- 453 FTP: File transfer incomplete, insufficient storage space.
- 500 Last command line completely unrecognized.
- 501 Syntax of last command in incorrect.
- 502 Last command incomplete, parameters missing.
- 503 Last command invalid (ignored), illegal parameter combination.
- 504 Last command invalid, action not possible at this time.
- 505 Last command conflicts illegally with previous command(s).
- 506 Requested action not implemented by the server.

Note

JFTP will only return the first two digits of any reply code.

On Unix, jftp returns:

- 0 on success.
- 1 on usage errors and so on
- the first two digits of the last FTP command result code when this is above 400.

Secure FTP

Use

To use SSL or TLS encrypted FTP connections in accordance with RFC 2218 you need a Secure Communications Library that is supplied by Redwood under license.

Once you have this, JFTP will adhere to the following RFCs:

- RFC 2228 [http://www.ietf.org/rfc/rfc2228.txt] - FTP Security Extensions.
- RFC 2246 [http://www.ietf.org/rfc/rfc2246.txt] - The TLS Protocol Version 1.0.

- RFC 4217 [http://www.ietf.org/rfc/rfc4217.txt] - Securing FTP with TLS.

Installation

Once you have obtained the license from Redwood and the secure communications library itself, put the shared library in the `bin` directory where `jftp` resides. This enables the following additional options:

Argument	Description
tls	Use a TLS v1 encrypted control connection.
ssl	Use an SSL v3 encrypted control connection. Redwood recommends that you use tls, instead.
protect	Use TLS/SSL for the data connection.
cert <file>	The location of the client certificate.
key <file>	The location of the client key.

Note

OpenSSL is required on each host where jftp is to be used.

jgetpar, jtool getpar

Use

A platform agent tool used to obtain configuration parameter values from within OS jobs.

This tool cannot be used outside job-context, so the `-j` or `--jobcontext` parameter is mandatory.

When an OS job runs on a platform agent it might be interested in the configuration parameters that the process server has sent to the platform agent. These are different from the parameters that the job definition has specified; these are always transmitted to the job using normal variables for the job definition type.

Syntax

```
jgetpar {-j | --jobcontext} <parameter>
```

Argument	Description
-j, --jobcontext	Run the command in job-context
<parameter>	Parameter name

Example

The following UNIX shell example shows how to determine the name of the process server.

```
echo This job is running for ProcessServer `jtool getpar -j ProcessServer`
```

jlink, jtool link

Use

A platform agent tool used to link files to jobs, so that they appear as output of the job in SAP CPS. You can then send these files via email from another, for example.

Syntax

```
jlink [--[no]deletewithjob] <file> [<shortname>]
```

Argument	Description
--deletewithjob	Determines if the file should be deleted with the job, the `no` prefix leaves the file intact after job deletion
<file>	The path to the file to attach to the job.
shortname	The short name used in SAP CPS

Note that the file should *not* itself reside in the `TMPDIR` directory, as all files in that directory will be deleted when the job finishes.

See The Environment of Platform Jobs for more information on the environment of jobs.

Example

Note

To perform the tasks below, you need to have the address of a valid mail server from which you are allowed to send email.

1. Create a job definition of type CMD with the below code, named *CMD_LinkFile* and leave all other job properties at their defaults.
2. Create a job chain named *JC_LinkFileAndSend*
   a. Add a Step, and a job running *CMD_LinkFile*
   b. Add a second step, and a job running *System_MailSend* with the below parameter values
3. Submit the job and check your mail.

Source for *CMD_LinkFile*

Note

you may change `%windir%\setuplog.txt` to an existing file on the file system, however, if you change `setup.log`, do not forget to adapt the below parameter.

```
rem Windows command job that attaches an external log file to a job with a short name
jlink --nodeletewithjob jlink %windir%\setuplog.txt setup.log
```

Parameters for the job of the second step of the job chain *JC_LinkFileAndSend*.

Parameter	Expression
To	<your email address>
Subject	Example file
Text	An example file sent to me by SAP CPS
Relative job	Step 1, Job 1
Job file	ntuser.ini
Server	<your SMTP server>

From	<your email address>

Note

Please replace <your email address> and <your SMTP server> with your real settings, ask your network administrator if in doubt.

jmail, jtool mail

Use

A platform agent tool used to send an email message.

This tool can be used outside job-context as a command line email client.

Syntax

```
jmail <options> <file> [<file> ... ]

  -server <server>: SMTP server address
  -port <portnr>:   Override port number, defaults to 25
  -o <sender>:      SMTP sender
  -i <from>:        'From:' address
  -t <to-list>:     'To:' recipient address list
  -c <cc-list>:     'Cc:' address list
  -b <bcc-list>:    'Bcc:' address list
  -s <subject>:     Subject
  -[no]mime:        whether to send as MIME message
  -mimetype <type>: MIME message type, for example 'text/plain' (default)
  -encoding:        MIME characterset, default 'ISO-8859-1'
  -a <file>:        MIME Attachment
  -secretfile:      Secretfile for SMTP Auth generated by jsecret
  -nosend           do everything but sending the actual email

  use - for file from stdin
```

Using Standard Input

When you specify `-a -`, you can provide the input from *standard input* by piping a file to jmail or typing the text out.

When you have finished typing, you can hit <Control+D> (hold down the `Ctrl` and the letter `d` keys, at the same time, starting with `Ctrl`) to terminate and send email.

Example

Send a mail with text file as body

```
jmail -server smtp.masalan.com -i jdoe@masalan.com -t jsmith@masalan.com -s "Please see attached" -a readme.t
```

Send a mail with an HTML file as body

```
jmail -server smtp.masalan.com -i jdoe@masalan.com -t jsmith@masalan.com -s "Please see attached" -mimetype t
```

Send a mail with an HTML body with an additional plain text file

```
jmail -server smtp.masalan.com -i jdoe@masalan.com -t jsmith@masalan.com -s "Please see attached" -mimetype t
```

Using standard in (-)

```
jmail -server smtp.masalan.com -i jdoe@masalan.com -t jsmith@masalan.com -s "Unimportant news" -a -
The cat caught the mouse,
the dog the cat,
the wolf the dog,
and the man the wolf.
<Control+D>
```

jmessage, jtool message

Use

A platform agent tool used to send an operator message.

This tool cannot be used outside job-context at the moment, so the `-j` or `--jobcontext` parameter is mandatory. As with all job-context operations, the update is performed asynchronously so there is no guarantee that the operation will succeed in the server.

As the operator message is logged in job-context, you will find that the operator message is linked to the job and can be accessed both from the job details and the operator message monitor.

Syntax

```
jmessage {-j | --jobcontext} <message>
```

Argument	Description
-j, --jobcontext	Run the command in job-context
<message>	Message text

Example

An example Microsoft Windows CMD script to show how you can raise awareness of errors beyond just logging them in job output files.

```
dir g:
if errorlevel 1 jmessage -j "Drive G: not accessible!"
```

jmonitor, jtool monitor

Use

A platform agent tool used to send monitor data values from within OS jobs.

This tool cannot be used outside job-context, so the `-j` or `--jobcontext` parameter is mandatory.

The Java server creates monitor nodes, checks and values automatically if needed.

The nodes are placed in the GLOBAL isolation group if they don't exist yet.

Customers that want to load data into other isolation groups should create the appropriate tree first, using the Java API.

Since a monitor value can only be set on a MonitorValue, and a MonitorValue needs to have a MonitorCheck as its parent node, and a MonitorCheck node must have a MonitorNode as its parent, the path that is passed needs to contain at least three levels, for example `/RootNode[/ChildNode...]/Check/Value`.

Syntax

```
jmonitor {-j | --jobcontext} <path>=<value> [<path>=<value> ...]
```

Argument	Description
-j, --jobcontext	Run the command in job-context
<path>	A monitor path
<value>	The new value for the monitor node.

Example

The following example shows the script source of a Perl script that runs on Microsoft Windows and UNIX systems with a GNU-like `df` binary. It finds out how much free space exists on the fixed disk drives / partitions and stores that data in the monitor tree.

```
use strict;

my $processServer = `jtool getpar -j ProcessServer -l info` or die "Cannot determine process server name";
$processServer =~ s/\s+$//;
my $cmd = "jtool monitor -j -l debug";
my $MonitorPath = " /System/ProcessServer/".$processServer."/Diskspace";

if ($^O eq "MSWin32")
{
  require Win32::OLE;

  my $server = Win32::NodeName();
  my $FIXED_DRIVE = 3;
  my $NETWORK_DRIVE = 4;

  my $locatorObj = Win32::OLE->new('WbemScripting.SWbemLocator') || die "Error creating locator object: ".Win
  $locatorObj->{Security_}->{impersonationlevel} = 3;
  my $serverObj = $locatorObj->ConnectServer($server,'root\cimv2',"","") || die "Error connecting to $server:

  printf "Free diskspace on $server:\n\n";
  printf "%-14s %-14s %-14s %-4s\n","Drive","Size","Free", "Perc";
  printf "%-14s %-14s %-14s %-4s\n","-" x 5,"-" x 14,"-" x 14, "-" x 4;

  foreach my $drive (Win32::OLE::Enum->All($serverObj->InstancesOf("Win32_LogicalDisk")))
  {
    next if ($drive->{DriveType} != $FIXED_DRIVE);

    my $dr = $drive->{DeviceID};
    my $fs = $drive->{FileSystem};
    my $sz = $drive->{Size};
    my $fb = $drive->{FreeSpace};
    my $perc = $fb / $sz * 100.0;
    printf "%-14s %-14.0f %-14.0f %-2.1f%%\n",$dr,$sz,$fb, $perc;
    $cmd .= $MonitorPath . "/" . $dr . "/Size=" . $sz;
    $cmd .= $MonitorPath . "/" . $dr . "/Free=" . $fb;
    $cmd .= $MonitorPath . "/" . $dr . "/PctFree=" . $perc;
  }
}
else
{
  # Assume UNIX with Gnu DF, for now
  open DF, "df -l -B1 |" or die "Cannot execute 'df': $!";
  printf "Free diskspace on `hostname`:\n\n";
  printf "%-14s %-14s %-14s %-4s\n","Drive","Size","Free", "Perc";
  printf "%-14s %-14s %-14s %-4s\n","-" x 14,"-" x 14,"-" x 14, "-" x 4;

  while (<DF>)
  {
    my ($fs, $sz, $us, $fb, $perc, $mount) = split /\s+/, $_;
    if ($mount =~ m/^\// && $sz > 1024 && $sz != $fb)
    {
      my $perc = $fb / $sz * 100.0;
      $mount =~ s/\//./g;
      $mount =~ s/\//./g;
      $mount =~ s/^\.//;
      $mount =~ s/^$/root/;
      printf "%-8s %-14.0f %-14.0f %-2.1f%%\n",$mount,$sz,$fb, $perc;
      $cmd .= $MonitorPath . "/" . $mount . "/Size=" . $sz;
      $cmd .= $MonitorPath . "/" . $mount . "/Free=" . $fb;
      $cmd .= $MonitorPath . "/" . $mount . "/PctFree=" . $perc;
    }
  }
  close DF;
}

print `$cmd`;
```

jregister, jtool register

Use

A platform agent tool used to register a platform agent with a central server so that only the registered SAP CPS instance can use the registered instance of the platform agent.

This is normally called by the installation procedure, and does not need to be called again. In some environments it could be useful for automated deployment of new platform agents.

Syntax

```
jregister <hostname> <port> <registration-filename> <secret-filename> [server_acl]
```

jsecret, jtool secret

Use

The jsecret platform agent tool creates an encrypted connection file which acts like password file, containing all the necessary information to connect to the central server, like username, password, hostname and port of the central server.

It can also be used to generate a shared secret, which is used by the platform agent and the central server to make sure the desired process server, from the desired central server, is connected to the desired platform agent.

Generating a Connection File

```
jsecret -c <filename> <hostname> <port> [<isolationGroup>]
```

Argument	Description
-c	Create a connection file
<filename>	The name of the file
<hostname>	The hostname of the target system
<port>	The port number of the target system
<IsolationGroup>	The isolation group to use

Generating a Shared Secret File

```
jsecret -n <filename>
```

Argument	Description	Default
-n	Create a shared secret	
<filename>	path to the file	net/instance/<instance_name>/private/secret

Example

Generating a connection file for registration

```
$ jsecret -c ./masalan.conn pr1.masalan.com 50000
Username:admin
Password:*****
$ ls -l ./masalan.conn
-rw------- 1 redwood users 240 nov 18 15:16 ./masalan.conn
```

Generating a connection file for jftp

```
$ jsecret -c ./ftp.sec ftp.masalan.com 21
Username:admin
Password:*****
$ ls -l ./ftp.sec
-rw------- 1 redwood users 240 nov 18 15:17 ./ftp.sec
```

Generating a shared secret file

```
jsecret -n net/instance/<instance>/private/secret
```

jscript, jtool script

Use

A platform agent tool used to execute RedwoodScript commands from the command line. This tool uses a connection file, and operation is synchronous: the script is executed the moment it receives an execute command.

Commands are always read from standard input. If standard input is a terminal and the silent option has not been set, a prompt will be shown. If standard input is a (redirected) file, pipe or stream, no prompt is shown. If you use the -s option, the only output given will be that sent by the server. This consists of user-created output via `jcsOut` and `jcsErr` as well as errors thrown by the interpreter (which go to stderr).

Syntax

```
jscript [-i] [-s] <connection-file>
```

Argument	Description
-i	Interactive, execution will not stop on successful execution of the first command.
-s	Silent, suppress prompts and feedback (but not error messages).
<connection file>	A file containing connection details, like host, port, username, password, isolation group.

Example

The following shows how we submit a job from a UNIX shell. In this example, the SAP CPS server is named `pr1.masalan.com`, its port is `50000` and the isolation group to use is `GLOBAL`. Please note that if you want to use the isolation group `GLOBAL`, you do not have to specify it as by default, the `GLOBAL` isolation group will be used. Note also, that when you enter your password for the connection file, no text is displayed as you type; this improves the security of your password.

```
$ jsecret -c ../../net/instance/default/masalan.conn pr1.masalan.com 50000 GLOBAL
Username:jdoe
Password:

$ jscript ../../net/instance/default/masalan.conn <<EOF
{
  JobDefinition jobDefinition = jcsSession.getJobDefinitionByName("System_Info");
  Job job = jobDefinition.prepare();
  jcsSession.persist();
}
/
EOF
```

If you perform the same but use the terminal to supply input you will see prompts:

```
$ jscript ../../net/instance/default/masalan.conn
Connected to masalan
jscript> {
      2>   JobDefinition jobDefinition = jcsSession.getJobDefinitionByName("System_Info");
      3>   Job job = jobDefinition.prepare();
      4>   jcsSession.persist();
      5> }
      6> /
Result: OK
$
```

jsleep, jtool sleep

Use

This platform agent tool is used to sleep a number of seconds. Not all platforms supported by SAP CPS have a built-in command to sleep a number of seconds, Microsoft Windows, for example.

Syntax

```
jsleep <seconds>
```

System Tools and Platform Agent Logging

Use

All tools, including the platform agent itself, log in a common logging format and use the same arguments to overrule the default logging settings.

These arguments are:

```
 [-f <file>] [-l <loglevel>]
```

Argument	Description

-f <file>	Filename where to log to. The default for tools is stderr.
-l <loglevel>	A single string that sets the log level. The default for the log-level is "info". The string is case-insensitive.

If the executable is called without the `-f` or `-l` switches the environment variables JCS_LOGFILE and JCS_LOGLEVEL are used. If these are not set then the default values are used.

Log-level

The log level parameter is defined as:

```
<loglevel> :== <element>,...
<element> :== [[<process>.]<category>=]<level>
<process> :== { network-processor | job-processor | jtool }
<level>   :== { none | fatal | error | warn | info | debug | net | trace }
```

- `<category>` can contain '*' to match any following characters
- `<category>` is one of a list of internal categories. To get an up-to-date explanation of all available variations of `<category>>` you can type `jtool -l help`.

Log File

The log file can contain a filename pattern, stderr, or - (for stderr). It may contain any of the following substitution parameters:

- ${ProcessId} - Process id of the process
- ${ProcessType} - Process type (standalone, service, ...)
- ${ProcessName} - Process name (network-processor, ...)
- ${InstallDirectory} - install directory.
- ${TraceDirectory} - ${DataRootDirectory}/trc.
- ${DataRootDirectory} - Base directory for platform agent files.
- ${ProcessServer} - Name of the process server that the agent is configured for.
- ${FileSeparator} - Character(s) between directory and file (for example `/` on Unix and `\` on Windows).

Logging in the system tools

The tools log to stderr, unless `-l <logfile>` is used or `JCS_LOGFILE` is set. The stderr output will end up in `stderr.log` when the tool is run as part of a scheduled job.

Logging in the platform-agent

The Process server parameters parameter `LogLevel` will override logging as soon as the Central Scheduler has established a connection with the Platform Agent. The default value for that parameter is `${TraceDirectory}${FileSeparator}${ProcessServer}-${ProcessName}-${ProcessId}.log.`

To analyze problems when the process server does not want to start, set the logging level as follows:

- UNIX: Add/change the log level in file `${InstallDir}/etc/startup/${instance}/loglevel` . The platform-agent script logs to `${InstallDir}/etc/log/<instance>/<instance>.log`. The network-processor logs to the same location until it is configured.
- Microsoft Windows: Change the settings for the instance using the *Scheduler Service Manager*; see the Configuring Platform Agents on Windows section for more information.

Example

To set a single environment variable that tells the network-processor to be very verbose, the job agent not at all, the jftp agent somewhat and the rest normal, you could use:

```
JCS_LOGLEVEL=info,jftp.*=warn,job-processor.*=error,network-processor.*=debug
```

Some more valid element examples:

```
Trace
*=DEBUG
http.*=info,job.*=debug
network-processor.*=warn
job-processor.*=debug
```

Creating AS/400 Process Servers

Use

This document covers the installation, syntax and operation of the AS/400 connector for SAP CPS. This can be used to manage jobs and file events on AS/400 systems. It does not cover installing SAP CPS to run on an application server (eg. SAP NetWeaver Web Application Server Java) on AS/400, that is covered by a separate document.

Terminology

We use the older names AS/400 and OS/400 to refer to the hardware and operating system as these are the names that have been in use longest, and correspond to older versions of the connector and the IBM Toolbox for Java naming. There are at least two other names for the machine:

From 2000-2004 the machine was known as the IBM eServer iSeries (generally abbreviated as iSeries).

Since 2004 the machine has been known as the IBM i5 and the operating system as i5/OS.

You can find more information about the history of the system on Wikipedia at http://en.wikipedia.org/wiki/AS/400.

All links to the IBM web site refer to i5/OS V5R4, if you are not using this version and are having problems with the syntax, check the exact syntax for your version.

Running Jobs

To run an AS/400 job SAP CPS makes a connection to the AS/400 using the AS/400 process server parameters `AS400Server, AS400User`, and `AS400Password` (the latter two can be overridden by the job definition Run As user). It then executes the command:

```
SBMJOB CMD(command) AS400JobAppend
```

This should produce a `CPC1221` message containing the AS/400 *job id*. This *job id* (and other information) is then written to the *Remote* attributes of the job, and can be seen in the job monitor.

The job is then monitored until it reaches any job status in *AS400JobTerminalStates*, or it reaches the job status `*ACTIVE` and any active status in *AS400JobFailOnActiveStatus*. During this process any messages on the job except those whose codes are listed in *AS400JobIgnoredExceptions* will cause the job to go into `Error`.

If the job reaches a status in *AS400JobTerminalStates*, then the spool files are retrieved according to the process server and job definition specification. If a job log is found, this log is parsed and the return code fetched. If the return code is not `0`, then the job is put into *Error*. If no log is found the job is put into *Completed.*

If it reaches a status in *AS400JobFailOnActiveStatus* then the job is put into *Error*.

Prerequisites

SAP CPS

- 0_24 or later.
- A license key containing ProcessServerServiceAS400.limit=<value>

AS/400

- OS/400 (aka i5 OS) V5R0 or later
  - Older versions are possible (down to V4R3)
  - Note: IBM only supports V5R1 up: https://www-304.ibm.com/systems/support/i/planning/software/i5os-schedule.html
- Connectivity from the host on which SAP CPS is installed to the AS/400
  - Ports 449, 446, 8470, 8471, 8472, 8473, 8474, 8475, 8476 need to be open.
  - See http://www-03.ibm.com/systems/i/software/toolbox/faqports.html

No additional software needs to be installed on the AS/400.

Procedure

Creating an AS/400 process server

You need one AS/400 process server for each managed AS/400 system.

Create the process server

To create an AS/400 process server:

1. Navigate to *Environment* > *Process Servers*.
2. Choose *New…* from the context menu.
3. Fill in the name and time zone.
4. On the *Job Definition Types* tab, fill in AS400 and add the job definition type.

5. On the *Queue Providers* tab, add a dedicated queue for the AS/400 system, we recommend a dedicated queue for each AS/400 with the same name as the AS/400. Other queues are optional.

6. On the process server parameters tab, add values for:

   a. *AS400Server* - If you don't need file events, and want to put the username/password in the job definition *Run As User* field.

   b. *AS400Server, AS400User, AS400Password* - If you want file events, or don't want to put the username/password in the job definition *Run As User* field.

See the AS/400 Process Server Parameters for more information on AS/400-related process server parameters.

Check that the process server is running:

1. Navigate to *Environment > Process Servers*.

2. Check the status icon of the process server, a *connecting* status can indicate a problem if it persists after 1 minute, refresh the view.

3. Expand the process server, check the status icon next to the AS400 service.

You can now start the process server. If it won't start, check the operator messages.

1. Navigate to *Mnitoring > Operator Messages*.

2. Scroll down to the latest entries.

3. Look for the name of the AS/400 process server.

Test the process server by running a job that should complete successfully

1. Create a job definition called *AS400_DSPJOB* containing only the text `DSPJOB`, and submit it to run on the AS/400 system. The job should run on the AS/400 and the output of the `DSPJOB` command should be visible in the *QPDSPJOB.1* job file. You should see the messages from the job turn up as job notes, and an operator message containing the return code.

Test the process server by running a job that should fail

1. Create another job definition called *AS400_Always_Fail*, with the command text below. This command will fail with an error code.

2. Check that the job return code and the return code in the job log match up.

   a. Navigate to *Monitoring > Jobs*

   b. locate the job that failed, choose it.

   c. In the lower details pane, expand *Files* and open `stdout` and `stdlog`, make sure the return code in the file matches the return code in the lower detail pane.

Command text for *AS400_Always_Fail*

```
DSPJOB JOB(000000/*N/REDWOOD)
```

Files on AS/400 Raise Events

Use

AS/400 file events are created in the same way as other file events. To check for files that match AS/400 file events, SAP CPS makes a connection to the AS/400 for each file event, using the AS/400 process server parameters `AS400Server, AS400User`, and `AS400Password`.

The connector then checks for files according to the directory and pattern specified by the file event. If a file is found it is optionally moved (if this is specified on the file event) and a file event is raised.

There are some differences:

- AS/400 file events are on the remote AS/400, which does not need an agent present, we connect over the network.
- The AS/400 file system does not work in the same way that a UNIX or Windows file system does.

For more information on the AS/400 file system, see the resources section.

Files in the library file system

The library file system provides you with access to database files and all of the other i5/OS object types that libraries can contain (http://publib.boulder.ibm.com/infocenter/iseries/v5r4/index.jsp?topic=/ifs/rzaaxlibfs.htm).

Files are often located in libraries, and libraries may be located inside other libraries (like directories). If the file you are looking for is in a library, you need to find the full path to the file you are looking for.

If the file you are looking for is inside a library (check with the operator), you will need to find the full path to it. On the AS/400, issue the `WRKLIB` command, and find the library you are looking for files in, it may be in another library (typically `QSYS` or `QGPL`). Once you have found the library you need to add `.LIB` to the end of each of the libraries, and separate them with /.

For example, if you are looking for files called `P*` in the `BEAVERS1` library, inside the `QSYS` library the file event should be specified as

```
Directory: /QSYS.LIB/AEVERS.LIB
Pattern: P*
```

Files in the root (/) file system

Files in this filesystem behave much like UNIX or Windows files, it has an hierarchical structure that users of these systems will be familiar with. (http://publib.boulder.ibm.com/infocenter/iseries/v5r4/index.jsp?topic=/ifs/rzaaxrootfs.htm).

Creating chains to manage the deletion of flag files

If you cannot use a move directory, you will need to use a job chain and a job definition to delete the file that triggers the event. To detect and remove files, create a job chain with a wait event that calls the job that should be submitted when the file arrives. In the second step of this chain, use a job definition that uses the `file.delete` action to delete the file after it has arrived.

Chapter 11. Reference

SAP RFC User Privileges

Use

Each ABAP stack you want to interact with needs a privileged user. SAP recommends it to be a System user for normal batch processing and a Dialog user if this user also has to be used as the Step user.

The following section describes the privileges required by the RFC user to interact with the ABAP stack of SAP Systems. To assign the following privileges, navigate to *Tools* > *Administration* > *User Maintenance* > *Role Administration* > *Roles* (transaction **PFCG**) and see the Assigning SAP Authorizations to the RFC User procedure for more information.

Note

Whenever an authorization problem occurs, you can log into the SAP system as the user and execute transaction **SU53**. You can use the output of this transaction to identify any missing authorizations.

SAP Authorizations for XBP and BW

AAAB - Cross-application Authorization Objects

S_RFC - Authorization check for RFC access

Name	**Required Authorizations**
Activity	* (or `Execute`)
Name of RFC to be protected	*, or all of BATG, FRFC, OCSB, RFC1, SALX, SCCA, SDIFRUNTIME, SDTX, SG00, SRFC, SXBP, SXMI, SYST, SVAR_RFC, SXBP_VAR, and SYSU
Type of RFC object to be protected	FUGR (Function group)

If you are using XBP transports, the following two RFC's need to be added:

/SCJS/1XBP, /SCJS/2XBP

If you are using ISU transports, the following two RFC's need to be added:

/SCJS/1ISU and /SCJS/2ISU

If you want to retrieve the SAP syslog from within SAP CPS with the GetSupportFiles functionality, you also need the following RFC:

SXMB

For BW, the list with names of RFCs to be protected has to be extended with following authorizations (unless the list contains just * (all RFCs)):

Name	**Required Authorizations**
Name of RFC to be protected	RSBC, RSAB, BATG, RSPC_API

This is required to be able to use RFC, and is thus an absolute requirement.

BC_A - Basis: Administration

S_ADMI_FCD - System Authorizations

Name	Required Authorizations
System administration functions	SP01, SP0R, SPAD

S_BTCH_ADM - Background Processing: Background Administrator

Name	Required Authorizations
Background administrator ID	*

S_BTCH_JOB - Background Processing: Operations on Background Jobs

Name	Required Authorizations
Job operations	*
Summary of jobs for a group	*

While it is possible to individually assign authorizations to delete background jobs, display spool requests, copy or repeat jobs, display the job processing log, release jobs and to display the job queue, all of them are required for proper function of the product.

S_BTCH_NAM - Background Processing: Background User Name

Name	Required Authorizations
Background User Name for Authorization	*

S_PROGRAM - ABAP: program run checks

Name	Required Authorizations
Authorization group ABAP program	* (or the required authorization group)
User action ABAP program	BTCSubmit

S_RZL_ADM - CCMS: System Administration

Name	Required Authorizations
Activity	01

S_SPO_ACT - Spool: Actions

Name	Required Authorizations
Authorization field for spool	*
Value for authorization check	*

S_SPO_DEV - Spool: Device authorizations

Name	Required Authorizations
Long device names	*

S_TABU_DIS - Table maintenance (via standard tools such as SM30)

Name	Required Authorizations
Activity	03
Authorization group	*

The *S_TABU_DIS* authorization is needed for importing BW InfoPackage groups. Additionally, it is required for all SAP releases that have neither XBP 3.0 nor transports (as of M28) in order to be able to import SAP calendars.

The following table illustrates the various combinations and the requirements:

	M28 (without transports) or earlier with XBP 2.0 or earlier	M28 (without transports) or earlier with XBP 3.0	M28 with transports
Run InfoPackagestable `RSMONRQTAB`	o	o	o
Import InfoPackage Groupstable `RSPAKPOS`	x	x	x
Import SAP Calendarstables `THOCS` and `TFACS`	x	-	-

- `o` - (optional) the official API will be used, which is slower and sometimes not reliable
- `x` - (mandatory) this functionality requires access to the table via `RFC_READ_TABLE`
- `-` - no direct table access is needed

S_OC_DOC

Only required if you want to archive spool lists.

Name	Required Authorizations
Activity	24 (or *)

S_OC_ROLE

Required for sending spool-lists to recipients or archiving spool lists

Name	Required Authorizations
Activity	ADMINISTRATOR (or *)

S_OC_SEND - Authorization Object for Sending

Required for sending spool lists to recipients

Name	Required Authorizations

Valid communication methods	* (or any communication methods you want to be able to use)
Range of number of recipients allowed per send operation	* (or the desired value, for example 100)

S_XMI_LOG - Internal access authorization for XMI log

Name	**Required Authorizations**
Access method for XMI log	*

S_XMI_LOG - Internal access authorization for XMI log

Name	**Required Authorizations**
Access method for XMI log	*

S_XMI_PROD - Auth. for external management interfaces (XMI)

Name	**Required Authorizations**
XMI logging: company name	REDWOOD (or *)
Product	*
Interface ID	*

Note

Please note that this has to be set to REDWOOD and not your company name.

This is the minimal set of authorizations required by SAP CPS.

SAP Authorizations for BW Process Chains

S_RS_ALL

You need to assign the S_RS_ALL profile to the user, this is done as follows:

If you want to schedule process chains and/or InfoPackages, then you must also assign the S_RS_ALL profile to the REDWOOD role. This can be done as follows:

1. Navigate to *Tools* > *Administration* > *User Maintenance* > *Role Administration* > *Roles* (transaction **PFCG**).

2. Create a new role `REDWOOD`, or edit the existing one if it already exists.

3. Select the *Authorizations* tab.

4. Choose *Change Authorization Data*. If the system shows a list of templates, choose *Do not select templates*.

5. You should now be in *Change role: Authorizations*.

6. Choose *Edit* > *Insert authorization(s)* > *From profile* and fill `S_RS_ALL` into the profile field, apply the change. Notice that the required authorizations have been added automatically.

S_DEVELOP - ABAP Workbench

When the *synchronous* flag is switched on, the following authorization is also required for process chains:

Name	Required Authorizations
ACTVT	16
DEVCLASS	*
OBJNAME	*
OBJTYPE	PROG
P_GROUP	*

AAAB - Cross-application Authorization Objects

SAP Authorizations required for XAL and XMW synchronization.

S_RFC - Authorization check for RFC access"

Name	Required Authorizations
Name of RFC to be protected	*, or all of FRFC, OCSB, SALX, SXMI, SYST, SDTX, RFC1, SDIFRUNTIME, SG00, SRFC, SYSU
Type of RFC object to be protected	FUGR

SAP Authorizations for Industry Solutions (ISU)

S_DEVELOP - ABAP Workbench

Name	Required Authorizations
ACTVT	03
DEVCLASS	EE20
OBJNAME	*
OBJTYPE	*
P_GROUP	*

SAP Authorizations for SAP Applications

The role *SAP_BC_REDWOOD_COMM_EXT_SDL* is required for Business Automation Enabler (BAE).

Please make sure, that the role has the following authorizations:

S_RFC_ADM

Name	Required Authorizations
Activity	All activities
Internet Communication Framework	*
Logical Destination	CRONACLE*, REDWOOD
Type of Entry in RFCDES	All values

Note

ABAP users connecting to SAP CPS require the role `SAP_BC_BATCH_ADMIN_REDWOOD`.

Default System Job Definitions

Use

System job definitions are internal job definitions and can be recognized by their prefix `System_` and their job definition type which is `System`. The following system job definitions are available:

- *System_Alert_Send* - Raise an ad hoc alert.
- *System_ArchiveAuditObjects* - Remove old audits.
- *System_ArchiveEvents* Remove cleared events.
- *System_Backup* - Detects database type for backup.
- *System_CreateMonitorTree* - Create a default monitor tree for all queues and process servers.
- *System_DeleteJobFiles* - Background job file deleter.
- *System_Derby_Backup* - Backup the Derby database to a file.
- *System_DynamicTrace* - Enable dynamic trace to provide additional information to support.
- *System_ExecuteExternalJob* - Runs a Redwood supplied external job.
- *System_ExplainJobStatus* - Explain why a job has a specific status.
- *System_Export_Application* - This job will export an application and all its descendent objects; they are exported to an archive that is attached to the job as a jobfile.
- *System_Export_RuleSet* - Export a rule-set.
- *System_ForceJobStatusUnknown* - Force running jobs to status *Unknown*.
- *System_GetSupportFiles* - Retrieve log files and other important SAP CPS information from a process server.
- *System_Get_Tutorials_List* - Retrieve the list of tutorials from www.redwood.com.
- *System_Ignored_Alert_Reporter* - Send reports for the number of ignored alerts by email to the addresses that have been ignored.
- *System_Import_Archive* - Import definitions from a CAR archive.
- *System_Import_JarFile* - Import a jar file into a library for use in SAP CPS.
- *System_Info* - Generate an overview of the system.
- *System_Isolation_Deisolation* - De-isolate the system by removing all isolation groups.
- *System_Isolation_SetUser* - Set a user to be able to read all isolation groups.
- *System_Mail_Configure* - Set (or detect) the mail server and from address to use.

- *System_Mail_Send* - Send an email using SMTP.
- *System_OperatorMessage* - Raise an operator message.
- *System_ProcessKeepClauses* - Background job definition keep clause processor.
- *System_RaiseClearEvent* - Allows to choose an EventDefinition and perform a clear or raise on it.
- *System_ReconfigureLogging* - Centrally reset the logging configuration.
- *System_RemoveDeletedJobFiles* - Remove deleted job files.
- *System_RemoveOldJobs* - Remove old jobs.
- *System_RemoveRepliedMessages* - Remove operator messages.
- *System_ReportRun* - Create a report based on a query filter.
- *System_Restore_Object* - Restore an object according to the audit.
- *System_Restore_Point_In_Time* - Restore objects or the entire repository to a point in time according to the audit.
- *System_SetDefaultLanguage* - Set Default Language.
- *System_SetSystemDataRootDirectory* - Set the DataRootDirectory of the System process server.
- *System_Sleep* - Sleep a number of seconds.
- *System_SMS_Configure* - Set the SMS/SMTP gateway to use.
- *System_SMS_Send* - Send an SMS using SMTP.
- *System_SNMP_SendTrap* - Send an SNMP trap.
- *System_UpdateJobStatistics* - Background job definition statistics updater.
- *System_Upgrade* - Perform any long running actions required after an upgrade.
- *System_Version_Info* - Generate an overview of the systems version and patch levels.

System_Alert_Send

Purpose

Raise an ad hoc alert, used in job chains to raise alerts in specific circumstances. This job definition requires the *Active Monitoring Module*.

Parameters

Parameter	**Direction**	**Description**	**Data type**
alertSource	Alert Source	In	String
data	Data	In	String
reply	Reply	Out	String

Schedule

This job definition is designed to be scheduled in job chains to raise alerts.

System_ArchiveAuditObjects

Purpose

Archive old audits, for future reference.

Parameters

Parameter	Direction	Description	Data type
AuditAgeUnitType	Age unit type	In	String
AuditAgeUnits	Age is older than	In	Number
BusinessKeyPattern	Business Key contains	In	String
CreationDate	Date is before	In	DateTimeZone
Kind	Archive audit objects by	In	String
ObjectType	Object Type	In	String

System_ArchiveEvents

Purpose

Archive old events, for future reference.

Parameters

Parameter	Direction	Description	Data type
ClearedDate	Remove before date:	In	DateTimeZone
EventAgeUnitType	Use age unit type:	In	String
EventAgeUnits	Remove before age:	In	Number
EventPrefix	Restrict search using event name prefix	In	String
Kind	Remove cleared events using:	In	String
RaisedSequence	Remove before raised sequence no:	In	Number

Schedule

This job can be scheduled regularly to decrease the amount of events stored in the repository.

System_Backup

Purpose

Perform a database backup. This is the database independent job definition that will submit the correct database backup script.

This job definition can be used to backup the database when no other third-party measures have already been taken to backup the database.

Note

When SAP CPS is running inside SAP NetWeaver AS Java, then there should already be a database backup planned for the SAP NetWeaver database, which SAP CPS uses as well.

Parameters

Parameter	Direction	Description	Data type
Filename	Filename	In	String
Platform	Platform	In	String

The Filename parameter can contain the following substitution parameters

- ${SystemId} - system id with all characters except A-Z, a-z and 0-9 replaced with underscores.
- ${date:format} - date using format 'format'. The format is any valid Java SimpleDateFormat.

Schedule

This job definition can be used to backup the database, on SAP NetWeaver, the database should already have a backup scheme, so this jobs definition does not need to be used.

System_CreateMonitorTree

Purpose

Was used to create the monitor tree that contains runtime information regarding process servers and queues; the tree is now generated and kept up-to-date automatically.

System_DeleteJobFiles

Purpose

Delete job output files of jobs that have been deleted. This job definition is automatically scheduled by the system, there is no need to run it manually.

Schedule

This job definition gets submitted every 15 min automatically, to customize this behavior, you can edit the submit frame `System_DeleteJobFilesSubmitFrame` in *Environment > Submit Frames*.

System_Derby_Backup

Purpose

This job definition is used to backup Apache Derby databases. It is submitted by the System_Backup job.

Parameter

Parameter	Description	Data Type
Filename	Filename	String

The Filename parameter can contain the following substitution parameters

- ${SystemId} - system id with all characters except A-Z, a-z and 0-9 replaced with underscores.
- ${date:format} - date using format 'format'. The format is any valid Java SimpleDateFormat.

System_DynamicTrace

Purpose

Enables dynamic tracing to help with problem solving. The format of the trace string is <category>=<level>;<category>=<level>;.... Valid categories are Redwood Java classes, like com.redwood.scheduler.; valid levels are debug, warn, info, error, and fatal. You should only use this job definition when asked to by SAP support staff, in which case they will provide you with an appropriate class and debug level.

Parameters

This job definition has one compulsory parameter:

Parameter	Description	Use	Data Type
TraceString	Trace string	Categories to be traced with verbosity level.	String

System_ExecuteExternalJob

Purpose

This job definition executes an external job definition (for example a patch or diagnostic tool) from a file. You should only use this job definition when asked to by SAP support staff, in which case they will provide you with an appropriate file.

Parameter	Description	Data Type
fileName	File name.	String

System_ExplainJobStatus

Purpose

This job definition is used to troubleshoot a job that remains in status `Queued`, it will provide an explanation as to why that job is in the `Queued` state.

Parameters

This job definition has one compulsory parameter:

Parameter	Description	Use	Data Type

jobId	Job Id.	The job id of the job to be examined.	Number

System_Export_Application

Purpose

This job will export an application and all its descendant objects; they are exported to an archive that is attached to the job as a job file.

Requires a *Import/Export Module* license, please check your license.

Parameters

Parameter	Description	Use	Data Type
applicationPath	Path of application to export.	The path to the application, For example *Redwood* for the *Redwood* application or *Redwood.System* for the *System* application inside the *Redwood* application.	String

System_Export_RuleSet

Purpose

Export various objects defined in an export rule-set. Used to schedule the export of specific objects.

Requires a *Import/Export Module* license, please check your license.

Parameters

Parameter	Description	Direction	Data type
exportRuleSetName	Name of the export rule-set which specifies the objects to export.	In	String

System_ForceJobStatusUnknown

Purpose

Force a running job, which should not or cannot be killed, into status *unknown*.

Parameters

Parameter	Description	Direction	Data type
JobId	Job ID	In	Number
ProcessServer	Process server	In	String

System_GetSupportFiles

Purpose

Retrieve the logs from a process server, operator messages and other important system information for troubleshooting. This job is automatically submitted when you use the *Get Support Files* action on a process server or a job that reached *Error*, for example ; it should not be submitted manually.

This information will be required by SAP support personnel when you want to report a problem.

Parameters

Parameter	Description	Direction	Data type
JobId	Job ID	In	Number
ProcessServerId	Process server id	In	String
Files	The list of files to retrieve	In	String

System_Get_Tutorials_List

Purpose

This job definition retrieves the list of tutorials that are available on www.redwood.com. These tutorials illustrate basic functionality and are free to view, a tutorial button will appear in the user interface when a tutorial is available for the object you are using. Some tutorials require you to register; registration is free.

Schedule

This job definition should be scheduled when you want to update the list of tutorials that are available.

System_Ignored_Alert_Reporter

Purpose

Send reports for the number of ignored alerts by email to the addresses that have been ignored. An address will be ignored if the number of emails sent to the address exceeds the alert rate limit (in an alert rate window). This job definition requires the Active Monitoring Module license.

Schedule

This job definition gets submitted every 15 min automatically, to customize this behavior, you can edit the submit frame `System_Ignored_Alert_ReporterSubmitFrame` in *Environment* > *Submit Frames*.

System_Import_Archive

Purpose

This job definition is used to import CAR archive files.

Requires a *Import/Export Module* license, please check your license.

Parameters

Parameter	Description	Data type
CarUrl	Filename or URL of archive, for example `R:\Car\Masalan.car` or `http://filserver.masalan.com/shared/CarFiles/Masalan.car`.	String
ImportRuleSetName	Name of import rule-set to apply to imported objects	String

System_Import_JarFile

Purpose

This job definition allows you to import all your Java packages from a JAR file into a library. It is called when you choose *Import JAR* from the context menu of a library.

Parameter	Description	Direction	Data type
Jar	Filename or URL of the archive	In	String
Library	Name of the library that the archive is to be a part of	In	String
auditReason	Reason for change	In	String

System_Info

Purpose

This job definition is used to retrieve information about the Java Virtual Machine that the System scheduler is running in. You may be asked to run this job by SAP Support for diagnostic purposes, there is no need to run it yourself.

System_Isolation_Deisolation

Purpose

This job definition is used to remove all isolation groups, allowing you to interact with more than one SAP system in a session, this requires the Chargeable version license.

System_Isolation_SetUser

Purpose

This job definition allows you to authorize a user to read all isolation groups.

Parameters

Parameter	Description	Use	Data Type
UserName	User Name	String	

System_Mail_Configure

Purpose

Configure the Mail module to allow the process server to send emails.

A Mail module license is necessary, please check your license.

Parameters

This job definition has two compulsory parameters:

Parameter	Description	Use	Data Type
From	From	The default *From:* Address used in the emails the process server sends.	String
Server	Server	The SMTP server to use	String

System_Mail_Send

Purpose

Send job output via email using SMTP.

A Mail module license is necessary, please check your license.

Parameters

Parameter	Description	Use	Data Type
From	From	The sender's email address.	String
JobFile	Job file	The output file of the *RelativeJob*. You can use regular expression to match multiple output files, `.*` matches all files.	String
RelativeJob	Relative job	The job that generated the output that is to be sent, syntax "job n', where *n* is the job id, for example *job 122588*. In a job chain, you can use the syntax *Step n, Job m* or *Step 2, Job 3* for the third job of the second step; it has to be a previous step to the one this job definition is in.	String
Server	Server	The SMTP server to use.	String
Subject	Subject	The subject of the email.	String
Text	Text	The text of the email.	String
To	To	The recipients' email address.	String

System_OperatorMessage

Purpose

This job definition is used to create Operator messages. You can use this job within a job chain to send one-way messages, or to send messages that wait for a reply. Use this job definition if you want to log that something has happened, or contact the operator to confirm a condition or provide input.

Parameters

Parameter	Description	Use	Data Type
message	Message.	The message that the operator has to reply to.	String
replyExpression	Reply Expression.	A regular expression of valid replies.	String
reply	Reply.	The reply to the operator message (output parameter)	String

System_ProcessKeepClauses

Purpose

Delete old jobs according to their keep clause. This job definition is automatically scheduled by the system, there is no need to run it manually.

Schedule

This job definition gets submitted every 15 min automatically, to customize this behavior, you can edit the submit frame `System_ProcessKeepClausesSubmitFrame` in *Environment* > *Submit Frames*.

System_RaiseClearEvent

Purpose

Allows to choose an EventDefinition and perform a clear or raise on it.

Parameters

Parameter	Description	Direction	Data type	Values
Action	Action	In	String	R (Raise) C (Clear)
EventDefinition	Event Definition	In	String	
Comment	Comment	In	String	

System_ReconfigureLogging

Purpose

Centrally reset the logging configuration.

System_RemoveDeletedJobFiles

Purpose

This job definition allows you to delete multiple old job files.

Parameters

Parameter	Description	Direction	Data type
ProcessServer	Process server	In	String
Code	Code	In	String

System_RemoveOldJobs

Purpose

Allows you to delete multiple jobs from the repository.

Note

Please do not schedule the job definition regularly, use keep clauses instead.

Parameters

Parameter	Description	Direction	Data type	Default Value	Default List of Values
JobAgeUnitType	Age unit type	In	String		
JobAgeUnits	Age is older than	In	Number		
JobDate	Date is before	In	DateTimeZone		
JobDefinitionPrefix	Filter by job definition name	In	String		
JobDefinition	The description of the job definition	In	Number		
JobId	JobId is before	In	Number		
Kind	Remove jobs by	In	String		I (JobId) D (Date) A (Age)
Statuses	Restrict to statuses (comma separated codes)	In	String	C,E,U,A	

System_RemoveRepliedMessages

Purpose

Removed old operator messages; messages that require a reply get a reply to before removal. This job definition is used to r

educe the number of operator messages in the operator message monitor.

Parameters

Parameter	Description	Direction	Data type	Default Value	Default List of Values
AutoReply	Automatically reply to messages?	In	String	true	true false

ReplyList	Comma separated list of valid replies	In	String		
Kind	Age or Date	In	String	D	D (date)A (age)
OperatorMessageDate	Date	In	Date Time Zone		
OperatorMessageAgeUnits	Age	In	Number		
OperatorMessageAgeUnitType	Age Units	In	String	S (seconds) N (minutes) H (hours) D (days) M (months) Y (years)	

System_ReportRun

Purpose

Generates a report on the resultset of a query filter, defined in a report object. The output can be generated in three formats, comma-separated, HTML, and XML.

Parameters

Parameter	Description	Direction	Data Type
OutputFormat	Report format	In	String
RecordCount	Maximum umber of records	Out	Number
Report	Report	In	String

System_Restore_Object

Purpose

Allows you to restore an object from the auditing record for the object at a point in time.

Note

This may have dependency issues if a dependent object has been deleted.

Parameters

Parameter	Description	Direction	Data type
BusinessKey	Business Key	In	String
DateTimeToRestore	Restoration Time	In	DateTimeZone
ObjectType	Object Type	In	String

System_Restore_Point_In_Time

Purpose

Allows you to restore the repository to a given point in time according to auditing.

Held jobs are generated to restore the system to a point in time. To continue with the restoration run the held jobs (in the order required to meet dependency requirements).

Parameter

Parameter	Description	Data Type
DateTimeToRestore	The point to restore to.	DateTimeZone

System_SetDefaultLanguage

Purpose

Set Default Language.

Parameters

Parameter	Description	Direction	Data type
Language	Language	In	String

System_SetSystemDataRootDirectory

Purpose

Set the DataRootDirectory of the System process server.

Parameters

Parameter	Description	Direction	Data type
Directory	Directory	In	String

System_Sleep

Purpose

Wait a number of seconds; allows to make your process wait for some seconds.

Parameter	Description	Data Type	Default Value
MilliSeconds	Time to sleep (in milliseconds)	Number	1000

System_SMS_Configure

Purpose

Set the SMS/SMTP gateway to use. The *SMS gateway* will be automatically detected using an email address of one of the mobile phones.

A Mail module license is necessary, please check your license.

Parameters

Parameter	Description	Direction	Data type
SMSGatewayPassword	The gateway password.	In	String
SMSGatewayUser	The gateway username.	In	String
SMSGateway	The gateway email address.	In	String

System_SMS_Send

Purpose

Send an SMS using SMTP.

A Mail module license is necessary, please check your license.

Parameters

Parameter	Description	Direction	Data type
MobileNumber	Mobile number	In	Number
Tags	Vendor Specific Tags	In	String
Text	Message	In	String

System_SNMP_SendTrap

Purpose

Send an SNMP trap.

A SNMP module license is necessary, please check your license.

Parameters

Parameter	Description	Direction	Data type
community	Community	In	String
data	Data	In	String
host	Host	In	String
id	Id	In	Number
port	Port	In	Number

System_UpdateJobStatistics

Use

Update the statistics of job definitions with runtime data from jobs that have run, this data is available in the statistics section in the details pane of *Definitions* > *Job Definitions*. This job definition is automatically scheduled by the system, there is no need to run it manually.

Schedule

This job definition gets submitted every 15 min automatically, to customize this behavior, you can edit the submit frame `System_UpdateJobStatisticsSubmitFrame` in *Environment > Submit Frames*.

System_Upgrade

Purpose

This job definition performs post-upgrade tasks and should not be submitted by users, unless you are instructed to do so by SAP support.

Parameters

Parameter	**Description**	**Data Type**
Mode	Mode	String

System_Version_Info

Purpose

Retrieve version information of SAP CPS. The output is especially useful for support requests.

Default SAP Job Definitions

Use

SAP-specific job definitions can be recognized by their prefix `SAP_` and their job definition type which is `SAPR3` for most, the only exceptions are for Java Scheduler job definitions that have the `JXBP` job definition type. Some parameters have both long and short versions of their values. The long values are used in drop-down menus in the submit wizard. The long versions are more verbose than the short ones, however, the short version must be used in job chains, so both are listed here.

The following SAP job definitions are available:

- *SAP_AbapRun*
- *SAP_AbapRunPrint*
- *SAP_AbapRunPrintExt*
- *SAP_BW_GenerateInfoPackageGroup*
- *SAP_BW_ImportInfoPackageGroups*
- *SAP_BW_ImportInfoPackages*
- *SAP_BW_ImportProcessChainDefinition*
- *SAP_BW_ImportProcessChains*
- *SAP_BW_InfopackageRun*

- *SAP_BW_MaintainProcessChainDefinition*
- *SAP_BW_Process*
- *SAP_BW_ProcessChain*
- *SAP_BW_ProcessChainRun*
- *SAP_BW_RestartProcess*
- *SAP_BW_RestartProcessChain*
- *SAP_CcmsJob*
- *SAP_CcmsJobCheckStatus*
- *SAP_CcmsJobGetApplicationLog*
- *SAP_CcmsJobGetApplicationLogHandles*
- *SAP_CcmsJobGetLog*
- *SAP_CcmsJobGetSpool list*
- *SAP_CcmsJobGetStatistics*
- *SAP_CcmsJobGetStatus*
- *SAP_CcmsJobModifyHeader*
- *SAP_ClosingCockpit_AbapRun*
- *SAP_ClosingCockpit_ManualTask*
- *SAP_CopyVariant*
- *SAP_DeleteCcmsMonitors*
- *SAP_DeleteVariant*
- *SAP_ExternalCommandRun*
- *SAP_ExternalProgramRun*
- *SAP_GenerateCcmsJobDefinition*
- *SAP_GetBatchResources*
- *SAP_GetEventHistory*
- *SAP_GetInstalledComponents*
- *SAP_GetXmiLog*
- *SAP_ImportAbapParameterDefinition*
- *SAP_ImportAbapPrograms*
- *SAP_ImportApplicationServers*

- *SAP_ImportArchiveObjects*
- *SAP_ImportBatchEvents*
- *SAP_ImportCalendars*
- *SAP_ImportCcmsJobs*
- *SAP_ImportCcmsMonitors*
- *SAP_ImportExternalCommands*
- *SAP_ImportOutputDevices*
- *SAP_ImportRecipients*
- *SAP_ImportVariantDefinition*
- *SAP_ImportVariants*
- *SAP_Info*
- *SAP_InterceptedJob*
- *SAP_JXBP_GenerateJobDefinition*
- *SAP_JXBP_ImportJobDefinitions*
- *SAP_MA_CopyParameters*
- *SAP_MA_CreateParameters*
- *SAP_MA_DeleteDunningProposal*
- *SAP_MA_ImportMassActivities*
- *SAP_MA_ImportMassActivity*
- *SAP_MA_MassActivityRun*
- *SAP_MA_MassActivityTemplateRun*
- *SAP_MA_ResumeMassActivity*
- *SAP_MA_ShowMassActivity*
- *SAP_MA_StopMassActivity*
- *SAP_MaintainXbpOptions*
- *SAP_PI_GetChannelStatus*
- *SAP_PI_ImportChannels*
- *SAP_PI_StartChannel*
- *SAP_PI_StopChannel*
- *SAP_RaiseEvent*

- *SAP_ReleaseInterceptedJobs*
- *SAP_ReloadApplicationLinks*
- *SAP_RunTemplateJob*
- *SAP_SetXmiAuditLevel*
- *SAP_SolutionManager_AbapRun*
- *SAP_StoreVariant*
- *SAP_SynchronizeEventHistoryCriteria*
- *SAP_SynchronizeInterceptionCriteria*
- *SAP_SynchronizeTerminatedCcmsJobs*

SAP_AbapRun

Use

This job definition is used to run ABAP's via the XBP interface in SAP Systems.

This job definition works with every version of the XBP interface, yet it offers a limited set of print and archive parameters (only those that are available in XBP 1.0) if you do not load the Redwood Transport files.

Parameters

Name	**Description**	**Data Type**	**Default Expression**	**Default List of Values**
ABAP_PROGRAM_NAME	ABAP program	String		
ABAP_VARIANT_NAME	ABAP variant	String		
TEMPORARY_VARIANT	Create temporary variant?	String	N	Y,N
ARCHIVE_INFO	Archive info	String		
ARCHIVE_OBJECT	Document type	String		
ARCHIVE_SAPOBJECT	Archive SAP object	String		
CLIENT	SAP client	String		
DELETE_JOB	Delete job when finished?	String	N	Y,N
JOBCLASS	SAP job class	String	C	A,B,C
JOBCOUNT	SAP job count	String		
JOBNAME	SAP job name	String		
LANGUAGE	SAP language for the step	String		
PRINT_ARCMODE	Print/Archive mode	String	1 (Print only)	1 (Print only) 2 (Archive only) 3 (Print and Archive)
PRINT_BANNERPAGE	Selection cover sheet?	String	N	Y,N
PRINT_COPIES	# of copies	String	1	

PRINT_DESTIN	Printer	String		
PRINT_EXPIRATION	Expiration in days	String	8	0 (Do not Delete)1-8 days
PRINT_IMMEDIATE	Print immediately?	String	N	Y,N
PRINT_NUMCOLUMNS	Print columns per page	String		
PRINT_NUMLINES	Print lines per page	String		
PRINT_RECIPIENT	Spool list recipient	String		
PRINT_RELEASE	Release print job?	String	N	Y,N
PRINT_SAPBANNER	SAP cover page?	String	D (System Default)	D (System Default)Y (Print)N (Do not print)
PRINT_SHOWPASSWD	Authority for printing	String		
SAP_SYSTEMS	SAP System	String		
SAP_USER_NAME	SAP user name	String		
SHOWLOG	Retrieve job log?	String	Y	Y,N
SHOWSPOOL	Retrieve spool list?	String	Y	Y,N
TARGET_GROUP	SAP target server group name	String		
TARGET_SERVER	SAP target server name	String		
TEMPORARY_VARIANT	Create temporary variant?	String	N	Y,N

If you set *Print/Archive mode* to *Archive only* or *Print and Archive*, the following additional Archive parameters must be filled:

- *Archive SAP object* - Referred to as *Obj.type* in SAPGUI
- *Archive object* - Referred to as *Doc.type* in SAPGUI
- *Archive info* - Referred to as *Information* in SAPGUI
- *Archive text information field* - Referred to as *Text* in SAPGUI

Note

If you need to use the optional *Archive text information field* parameter, known as *Text* in SAPGUI, you will have to use the job definition `SAP_AbapRunPrint` or `SAP_AbapRunPrintExt`.

Advanced

By adding the following two parameters, you can make jobs wait on events in SAP:

Parameter	**Description**	**Direction**	**Data Type**
STARTCOND_EVENTID	batch event id	In	String
STARTCOND_EVENTPARM	parameter of the event id	In	String

Note that you have to duplicate the job definition and modify the duplicate.

More Information

Handling ABAP programs

SAP_AbapRunPrint

Use

This job definition is used to run ABAP's via the XBP interface in SAP Systems.

This job definition makes use of the extended capabilities of the XBP 2.0 interface, which provides an extended set of print and archive parameters.

This job definition works on every version of the XBP interface, yet on versions below 2.0 the new functionality is disabled if you do not have the transport files installed.

Parameters

Name	**Description**	**Data Type**	**Default Expression**	**Default List of Values**
ABAP_PROGRAM_NAME	ABAP program	String		
ABAP_VARIANT_NAME	ABAP variant	String		
TEMPORARY_VARIANT	Create temporary variant?	String	N	Y,N
ARCHIVE_ARCTEXT	Archive text information field	String		
ARCHIVE_AR_OBJECT	Document type	String		
ARCHIVE_INFO	Archive info	String		
ARCHIVE_SAP_OBJECT	Archive SAP object	String		
CLIENT	SAP client	String		
DELETE_JOB	Delete job when finished?	String	N	Y, N
JOBCLASS	SAP job class	String	C	A, B, C
JOBCOUNT	SAP job count	String		
JOBNAME	SAP job name	String		
LANGUAGE	SAP language for the step	String		
PRINT_ARMOD	Print/Archive mode	String	1	
PRINT_FOOTL	Print a footer?	String	N	Y, N
PRINT_LICT	Print lines per page	String		
PRINT_LISZ	Print columns per page	String		
PRINT_PAART	Print format	String		
PRINT_PDEST	Printer	String		
PRINT_PEXPI	Expiration in days	String	8	0 (Do not Delete)1-8 days
PRINT_PLIST	Spool list name	String		
PRINT_PRABT	Department on cover page	String		
PRINT_PRBER	Authority for printing	String		
PRINT_PRBIG	Selection cover sheet?	String	N	Y, N

PRINT_PRCOP	# of copies	String	1 (Print only)	1 (Print only) 2 (Archive only) 3 (Print and Archive)
PRINT_PRDSN	Spool dataset name	String		
PRINT_PRIMM	Print immediately?	String	N	Y, N
PRINT_PRNEW	New spool?	String	Y	Y, N
PRINT_PRREC	Spool list recipient	String		
PRINT_PRREL	Release print job?	String	N	Y, N
PRINT_PRSAP	SAP cover page?	String	D (System Default)	D (System Default)Y (Print)N (Do not print)
PRINT_PRTXT	Cover sheet title	String		
PRINT_PRUNX	OS cover sheet?	String	D (System Default)	D (System Default)Y (Print)N (Do not print)
PRINT_PTYPE	Spool type	String		
SAP_SYSTEMS	SAP System	String		
SAP_USER_NAME	SAP user name	String		
SHOWLOG	Retrieve job log?	String	Y	Y, N
SHOWSPOOL	Retrieve spool list?	String	Y	Y, N
TARGET_GROUP	SAP target server group name	String		
TARGET_SERVER	SAP target server name	String		
TEMPORARY_VARIANT	Create temporary variant?	String	N	Y,N

If you set *Print/Archive mode* to *Archive only* or *Print and Archive*, the following Archive parameters should be filled:

- *Archive SAP object* - Referred to as Obj.type in SAPGUI
- *Archive object* - Referred to as Doc.type in SAPGUI
- *Archive Info* - Referred to as Information in SAPGUI

Advanced

By adding the following two parameters, you can make jobs wait on events in SAP:

Parameter	**Description**	**Direction**	**Data Type**
STARTCOND_EVENTID	batch event id	In	String
STARTCOND_EVENTPARM	parameter of the event id	In	String

Note that you have to duplicate the job definition and modify the duplicate.

More Information

Handling ABAP programs

SAP_AbapRunPrintExt

Use

This job definition is used to run ABAP's via the XBP interface in SAP Systems.

This job definition makes use of the extended capabilities of the XBP 3.0 interface to set the spool list recipient and retrieve application log and batch statistics of the SAP job.

This job definition works on every version of the XBP interface, yet on versions below 3.0 the new functionality is disabled if you do not have the transport files installed.

Parameters

Name	Description	Data Type	Default Expression	Default List of Values
ABAP_PROGRAM_NAME	ABAP program	String		
ABAP_VARIANT_NAME	ABAP variant	String		
TEMPORARY_VARIANT	Create temporary variant?	String	N	Y,N
ARCHIVE_ARCTEXT	Archive text information field	String		
ARCHIVE_AR_OBJECT	Document type	String		
ARCHIVE_INFO	Archive info	String		
ARCHIVE_SAP_OBJECT	Archive SAP object	String		
CLIENT	SAP client	String		
DELETE_JOB	Delete job when finished?	String	N	
JOBCLASS	SAP job class	String	C	A, B, C
JOBCOUNT	SAP job count	String		
JOBNAME	SAP job name	String		
LANGUAGE	SAP language for the step	String		
PRINT_ARMOD	Print/Archive mode	String	1 (Print only)	1 (Print only) 2 (Archive only) 3 (Print and Archive)
PRINT_FOOTL	Print a footer?	String	N	Y, N
PRINT_LICT	Print lines per page	String		
PRINT_LISZ	Print columns per page	String		
PRINT_PAART	Print format	String		
PRINT_PDEST	Printer	String		
PRINT_PEXPI	Expiration in days	String	8	0 (Do not Delete)1-8 days
PRINT_PLIST	Spool list name	String		
PRINT_PRABT	Department on cover page	String		
PRINT_PRBER	Authority for printing	String		
PRINT_PRBIG	Selection cover sheet?	String	N	Y, N
PRINT_PRCOP	# of copies	String	1	

PRINT_PRDSN	Spool dataset name	String		
PRINT_PRIMM	Print immediately?	String	N	Y, N
PRINT_PRNEW	New spool?	String	Y	
PRINT_PRREC	Spool list recipient	String		
PRINT_PRREL	Release print job?	String	N	Y, N
PRINT_PRSAP	SAP cover page?	String	D (System Default)	D (System Default)Y (Print)N (Do not print)
PRINT_PRTXT	Cover sheet title	String		
PRINT_PRUNX	OS cover sheet?	String	D (System Default)	D (System Default)Y (Print)N (Do not print)
PRINT_PTYPE	Spool type	String		
REC_BLIND_COPY	Recipient blind copy flag	String	N	Y, N
REC_COPY	Recipient copy flag	String	N	Y, N
REC_DELIVER	Report send status	String	_ (System Default)	_ (System Default) A (Always) E (On error) N (Never)
REC_EXPRESS	Recipient express flag	String	N	Y, N
REC_MAILSTATUS	Report status by e-mail	String	_ (System Default)	_ (System Default) A (Always) E (On error) N (Never)
REC_NO_FORWARDING	Recipient no forwarding flag	String	N	Y, N
REC_NO_PRINT	Recipient no print flag	String	N	Y, N
REC_RECIPIENT	Spool list recipient	String		
REC_TYPE	Recipient type	String	U (Internet User)	U (Internet User) B (SAP user) O (SAP-Office user) P (Personnel distribution list) C (Shared distribution list) F (Fax number)
SAP_SYSTEMS	SAP System	String		
SAP_USER_NAME	SAP user name	String		
SHOWAPPLLOG	Retrieve application logs?	String	Y	Y, N
SHOWLOG	Retrieve job log?	String	Y	Y, N
SHOWSPOOL	Retrieve spool list?	String	Y	Y, N
SHOWSTAT	Retrieve job statistics?	String	Y	Y, N
TARGET_GROUP	SAP target server group name	String		
TARGET_SERVER	SAP target server name	String		
TEMPORARY_VARIANT	Create temporary variant?	String	N	Y,N

If you set *Print/Archive mode* to *Archive only* or *Print and Archive*, the following Archive parameters should be filled:

- *Archive SAP object* - Referred to as Obj.type in SAPGUI
- *Archive object* - Referred to as Doc.type in SAPGUI
- *Archive Info* - Referred to as Information in SAPGUI

Advanced

By adding the following two parameters, you can make jobs wait on events in SAP:

Parameter	Description	Direction	Data Type
STARTCOND_EVENTID	batch event id	In	String
STARTCOND_EVENTPARM	parameter of the event id	In	String

Note that you have to duplicate the job definition and modify the duplicate.

More Information

Handling ABAP programs

SAP_BW_GenerateInfoPackageGroup

Use

Generate an InfoPackage group in an SAP System.

Parameters

Name	Description	Direction	Data Type	Default Expression
SAP_SYSTEMS	SAP System	In	String	
INFOPACKAGEGROUP_NAME	SAP BW InfoPackage group	In	String	
JOBDEF_NAME	Name for the generated job definition	In	String	
OVERWRITE	Overwrite existing job definitions?	In	String	Y
APPLICATION	Application	In	String	
SHOWLOG	Retrieve job log?	In	String	Y

SAP_BW_ImportInfoPackageGroups

Use

Import InfoPackage groups from an SAP System.

Parameter

Name	Description	Direction	Data Type
SAP_SYSTEMS	SAP System	In	String

SAP_BW_ImportInfoPackages

Use

Import InfoPackages.

Parameters

Name	Description	Direction	Data Type	Default Expression
SAP_SYSTEMS	SAP System	In	String	
INFOPACKAGE_NAME	Technical name of the SAP BW InfoPackage	In	String	
MAX_COUNT	Max. count	In	Number	
DROP_OLD_SELECTION	Drop old selection?	In	String	Y
IPAK_INFOSOURCE	Info source of the SAP BW InfoPackage	In	String	
IPAK_SOURCESYSTEM	Source system of the SAP BW InfoPackage	In	String	
IPAK_DATASOURCE	Datasource of the SAP BW InfoPackage	In	String	

SAP_BW_ImportProcessChainDefinition

Use

Import the definition of a Process Chain.

Parameters

Name	Description	Direction	Data Type	Default Expression
SAP_SYSTEMS	SAP System	In	String	
PCHAIN_NAME	Technical name of the SAP BW Process Chain	In	String	
DROP_OLD_SELECTION	Drop old selection?	In	String	Y

More Information

Importing and Running Process Chains

SAP_BW_ImportProcessChains

Use

Import process chains.

Parameters

Name	Description	Direction	Data Type	Default Expression
DROP_OLD_SELECTION	Drop old matching values?	In	String	Y
IMPORT_CHAIN_DEFINITION	Import process chain definition?	In	String	
MAX_COUNT	Max. count	In	Number	
PCHAIN_NAME	Technical name of the SAP BW Process Chain	In	String	
PCHAIN_TEXT	Description of the SAP BW process chain	In	String	
SAP_SYSTEMS	SAP System	In	String	

More Information

Importing and Running Process Chains

SAP_BW_InfopackageRun

Use

This job definition is used to run InfoPackages in SAP BW.

Parameters

Name	Description	Data Type	Direction	Default Expression
JOBNAME	SAP BW job name	In	String	
NAME	Technical name of the SAP BW InfoPackage	In	String	
REQUESTID	Request ID of an SAP BW InfoPackage run	Out	String	
SAP_SYSTEMS	SAP System	In	String	
SHOWLOG	Retrieve job log?	In	String	Y

SAP_BW_MaintainProcessChainDefinition

Use

Maintain the definition of a Process Chain.

Parameters

Name	Description	Direction	Data Type	Default Expression	Default List of Values

SAP_SYSTEMS	SAP System	In	String		
PCHAIN_NAME	Technical name of the SAP BW Process Chain	In	String		
BW_PROCESS_TYPE	Type of the SAP BW process	In	String		
BW_PROCESS_VARIANT	Variant (name) of the SAP BW process	In	String		
BW_MAX_RESTARTS	Number of automatic restarts on error	In	Number		
BW_RESTART_DELAY	Delay needed to restart.	In	Number	10	
BW_RESTART_DELAY_TIME_UNIT	Unit of the delay (seconds or minutes)	In	String	M(Minutes)	M (Minutes) S (Seconds)
BW_RESTART_UNCHECKED	Restart unchecked	In	String		
BW_ERROR_PROCESS_TYPE	Type of the SAP BW process to restart on error	In	String		
BW_ERROR_PROCESS_VARIANT	Variant (name) of the SAP BW process to restart on error	In	String		

SAP_BW_Process

Use

This job definition represents a process of a BW Process Chain. It is not supposed to be scheduled by the user.

Parameters

Name	Description	Data Type	Default Expression
ID	SAP BW process id	String	
JOBCOUNT	SAP BW process batch job count	String	
LOGID	SAP BW process chain logid	String	
NAME	SAP BW process name	String	
SHOWLOG	Retrieve job log?	String	Y
TYPE	SAP BW process type	String	

SAP_BW_ProcessChain

Use

This job definition represents an SAP BW Process Chain. It is not supposed to be scheduled by the user.

Parameters

Name	Description	Direction	Data Type	Default Expression

SAP_SYSTEMS	SAP System	In	String	
NAME	Technical name of the SAP BW Process Chain	In	String	
LOGID	SAP BW process chain LogID	In/Out	String	
SYNCHRONOUS	Synchronous run?	In	String	N
SIMULATE	Simulate run?	In	String	N
NOPLAN	No plan?	In	String	N
SHOWLOG	Retrieve job log?	In	String	Y

SAP_BW_RestartProcess

Use

This job restarts an instance of a process of an SAP BW Process Chain. You do not have to submit this job definition, when a job of a process chain fails, the parent job of the job chain will have an operator message with which you can restart the process chain or one or more of its children. Restarting process chains requires *ProcessServerService.SAP.ProcessChainRestart* license key.

Parameters

Name	**Description**	**Direction**	**Data Type**	**Default Expression**	**Values**
SAP_SYSTEMS	SAP System	In	String		
NAME	SAP BW process name	In	String		
ID	SAP BW process id	In	String		
TYPE	SAP BW process type	In	String		CHAIN, TRIGGER, ABAP ...
PCHAIN_NAME	Technical name of the SAP BW Process Chain	In	String		
LOGID	SAP BW process chain LogID	In	String		
BW_RESTART_UNCHECKED	Restart unchecked	In	String	N	Y, N
JOBCOUNT	SAP BW process batch job count	In	String		
SHOWLOG	Retrieve job log?	In	String	N	Y, N

More Information

Restarting Process Chains

SAP_BW_RestartProcessChain

Use

This job restarts an instance of an SAP BW Process Chain. You do not have to submit this job definition, when a job of a process chain fails, the parent job of the job chain will have an operator message with which you can restart the process chain or one or more of its children. Restarting process chains requires *ProcessServerService.SAP.ProcessChainRestart* license key.

Parameters

Name	Description	Direction	Data Type	Default Expression
SAP_SYSTEMS	SAP System	In	String	
NAME	Technical name of the SAP BW Process Chain	In	String	
LOGID	SAP BW process chain LogID	In/Out	String	
BW_RESTART_UNCHECKED	Restart unchecked	In	String	Y

SAP_BW_ProcessChainRun

Use

This job definition is used to run BW Process Chains in SAP.

Parameters

Name	Description	Data Type	Default Expression	Default List of Values
CHAIN_RESTARTABLE	Make processes restartable on error?	String	F (Failed)	F (Failed) A (All) N (Not restartable)
LOGID	SAP BW process chain LogID	String		
NAME	Technical name of the SAP BW Process Chain	String		
NOPLAN	No plan?	String	N	Y, N
SAP_SYSTEMS	SAP System	String		
SHOWLOG	Retrieve job log?	String	Y	Y, N
SIMULATE	Simulate run?	String	N	Y, N
SYNCHRONOUS	Synchronous run?	String	N	Y, N

SAP_CcmsJob

Use

This job definition represents an SAP batch process that is monitored by SAP CPS yet running under full control of the SAP System. It is not supposed to be scheduled by the user.

Parameters

Name	Description	Data Type	Default Expression
CLIENT	SAP client	String	

DELETE_JOB	Delete job when finished?	String	N
JOBCOUNT	SAP job count	String	
JOBNAME	SAP job name	String	
SAP_USER_NAME	SAP user name	String	
SHOWLOG	Retrieve job log?	String	Y
SHOWSPOOL	Retrieve spool list?	String	Y

SAP_CcmsJobCheckStatus

Use

This job checks if the status of up to 10 SAP batch jobs in the database is valid and updates the database otherwise. It is not supposed to be scheduled by the user.

Parameters

Name	**Description**	**Direction**	**Data Type**
CLIENT	SAP client	In	String
JOBCOUNT	SAP job count	In	String
JOBCOUNT_1	SAP job count 1	In	String
JOBCOUNT_10	SAP job count 10	In	String
JOBCOUNT_2	SAP job count 2	In	String
JOBCOUNT_3	SAP job count 3	In	String
JOBCOUNT_4	SAP job count 4	In	String
JOBCOUNT_5	SAP job count 5	In	String
JOBCOUNT_6	SAP job count 6	In	String
JOBCOUNT_7	SAP job count 7	In	String
JOBCOUNT_8	SAP job count 8	In	String
JOBCOUNT_9	SAP job count 9	In	String
JOBNAME	SAP job name	In	String
JOBNAME_1	SAP job name 1	In	String
JOBNAME_10	SAP job name 10	In	String
JOBNAME_2	SAP job name 2	In	String
JOBNAME_3	SAP job name 3	In	String
JOBNAME_4	SAP job name 4	In	String
JOBNAME_5	SAP job name 5	In	String
JOBNAME_6	SAP job name 6	In	String
JOBNAME_7	SAP job name 7	In	String
JOBNAME_8	SAP job name 8	In	String
JOBNAME_9	SAP job name 9	In	String
JOB_STATUS	Job status	Out	String
JOB_STATUS_1	Job status 1	Out	String

JOB_STATUS_10	Job status 10	Out	String
JOB_STATUS_2	Job status 2	Out	String
JOB_STATUS_3	Job status 3	Out	String
JOB_STATUS_4	Job status 4	Out	String
JOB_STATUS_5	Job status 5	Out	String
JOB_STATUS_6	Job status 6	Out	String
JOB_STATUS_7	Job status 7	Out	String
JOB_STATUS_8	Job status 8	Out	String
JOB_STATUS_9	Job status 9	Out	String
SAP_SYSTEMS	SAP System	In	String

SAP_CcmsJobGetApplicationLog

Use

Get an application log of an SAP batch job.

Parameters

Name	**Description**	**Direction**	**Data Type**
SAP_SYSTEMS	SAP System	In	String
CLIENT	SAP client	In	String
JOBNAME	SAP job name	In	String
JOBCOUNT	SAP job count	In	String
JOB_ID	Job id	In	Number
LOGHANDLE	Application log handle	In	String

SAP_CcmsJobGetApplicationLogHandles

Use

Get the list of application log handles of an SAP batch job.

Parameters

Name	**Description**	**Direction**	**Data Type**
SAP_SYSTEMS	SAP System	In	String
CLIENT	SAP client	In	String
JOBNAME	SAP job name	In	String
JOBCOUNT	SAP job count	In	String
JOB_ID	Job id	In	Number

SAP_CcmsJobGetLog

Use

Get the job log of an SAP batch job.

Parameters

Name	Description	Direction	Data Type	Default Value
CLIENT	SAP client	In	String	
CONSIDER_SIZE_CONSTRAINTS	Consider file size constraints?	In	String	N
JOBCOUNT	SAP job count	In	String	
JOBNAME	SAP job name	In	String	
JOB_ID	Job id	In	Number	
SAP_SYSTEMS	SAP System	In	String	

SAP_CcmsJobGetSpool list

Use

Get a spool list of an SAP batch job.

Name	Description	Direction	Data Type	Default Value
CLIENT	SAP client	In	String	
CONSIDER_SIZE_CONSTRAINTS	Consider file size constraints?	In	String	N
JOBCOUNT	SAP job count	In	String	
JOBNAME	SAP job name	In	String	
JOB_ID	Job id	In	Number	
SAP_SYSTEMS	SAP System	In	String	
SPOOLID	Spool list ID	In	String	
STEP_NUMBER	Step number of the SAP batch job	In	Number	

SAP_CcmsJobGetStatistics

Use

Get the runtime statistics of an SAP batch job.

Parameters

Name	Description	Direction	Data Type

SAP_SYSTEMS	SAP System	In	String
CLIENT	SAP client	In	String
JOBNAME	SAP job name	In	String
JOBCOUNT	SAP job count	In	String
JOB_ID	Job id	In	Number

SAP_CcmsJobModifyHeader

Use

Parameters

Name	**Description**	**Direction**	**Data Type**	**Default Expression**
CLIENT	SAP client	In	String	
EVENT_ID	Event ID	In	String	
EVENT_PARAMETER	Event parameter	In	String	
JOBCOUNT	SAP job count	In	String	
JOBNAME	SAP job name	In	String	
PERIODIC	Periodic job?	In	String	
READ_JOB_INFO	Read job info?	In	String	Y
RELEASE_JOB	Release job in SAP?	In	String	N
SAP_SYSTEMS	SAP System	In	String	
START_TIME	Start time	In	DateTimeZone	
TARGET_GROUP	SAP target server group name	In	String	
TARGET_SERVER	SAP target server name	In	String	

SAP_CcmsJobGetStatus

Use

This job retrieves the status of an SAP batch job.

Parameters

Name	**Description**	**Direction**	**Data Type**
SAP_SYSTEMS	SAP System	In	String
CLIENT	SAP client	In	String
JOBNAME	SAP job name	In	String
JOBCOUNT	SAP job count	In	String
JOB_STATUS	Job status	Out	String

SAP_ClosingCockpit_AbapRun

Use

Run an ABAP report from SAP Closing Cockpit.

Parameters

Name	Description	Direction	Data Type	Default Expression	Default List of Values
SAP_SYSTEMS	SAP System	In	String		
CLIENT	SAP client	In	String		
JOBNAME	SAP job name	In	String		
ABAP_PROGRAM_NAME	ABAP program	In	String		
ABAP_VARIANT_NAME	ABAP variant	In	String		
TEMPORARY_VARIANT	Create temporary variant?	String	N	Y,N	
JOBCLASS	SAP job class	In	String	C	A, B, C
SAP_USER_NAME	SAP user name	In	String		
TARGET_SERVER	SAP target application server name	In	String		
JOBDOCID	Job documentation ID	In	String		
PRINT_ARMOD	Print/Archive mode	In	String	1 (Print only)	1 (Print only) 2 (Archive only) 3 (Print and Archive)
PRINT_PDEST	Printer	In	String		
PRINT_PRTXT	Cover sheet title	In	String		
PRINT_PRCOP	# of copies	In	String	1	
PRINT_PRNEW	New spool?	In	String	Y	Y, N
PRINT_PLIST	Spool list name	In	String		
PRINT_PRIMM	Print immediately?	In	String	N	Y, N
PRINT_PRREL	Release print job?	In	String	N	Y, N
PRINT_PEXPI	Expiration in days	In	String	8	0 (Do not Delete)1-8 days
PRINT_LICT	Print lines per page	In	String		
PRINT_LISZ	Print columns per page	In	String		
PRINT_PAART	Print format	In	String		
PRINT_PRBIG	Selection cover sheet?	In	String	N	Y, N
PRINT_PRSAP	SAP cover page?	In	String	D (System Default)	D (System Default)Y (Print)N (Do not print)
PRINT_PRREC	Spool list recipient	In	String		

PRINT_PRABT	Department on cover page	In	String		
PRINT_PRBER	Authority for printing	In	String		
PRINT_PRDSN	Spool dataset name	In	String		
PRINT_PTYPE	Spool type	In	String		
PRINT_FOOTL	Print a footer?	In	String	N	Y, N
PRINT_PRUNX	OS cover sheet?	In	String	D (System Default)	D (System Default)Y (Print)N (Do not print)
ARCHIVE_SAP_OBJECT	Archive SAP object	In	String		
ARCHIVE_AR_OBJECT	Document type	In	String		
ARCHIVE_INFO	Archive info	In	String		
ARCHIVE_ARCTEXT	Archive text information field	In	String		
REC_RECIPIENT	Spool list recipient	In	String		
REC_TYPE	Recipient type	In	String	U (Internet User) B (SAP user) O (SAP-Office user) P (Personnel distribution list) C (Shared distribution list) F (Fax number)	
REC_COPY	Recipient copy flag	In	String	N	Y, N
REC_BLIND_COPY	Recipient blind copy flag	In	String	N	Y, N
REC_EXPRESS	Recipient express flag	In	String	N	Y, N
REC_NO_FORWARDING	Recipient no forwarding flag	In	String	N	Y, N
REC_NO_PRINT	Recipient no print flag	In	String	N	Y, N
REC_DELIVER	Report send status	In	String	_ (System Default)	_ (System Default) A (Always) E (On error) N (Never)
REC_MAILSTATUS	Report status by e-mail	In	String	_ (System Default)	_ (System Default) A (Always) E (On error) N (Never)
SAP_CC_LDAY_P	Closing key day (as single value)	In	String		

SAP_CC_LDAY_S	Closing key day (as range)	In	String		
SAP_CC_PERIOD_P	Period (as single value)	In	String		
SAP_CC_PERIOD_S	Period (as range)	In	String		
SAP_CC_GJAHR_P	Financial year (as single value)	In	String		
SAP_CC_GJAHR_S	Financial year (as range)	In	String		
SAP_CC_RESPONSIBLE	Responsible	In	String		
SAP_CC_EXECUTIVE	Executioner	In	String		
SAP_CC_ESTIMATED_RUNTIME	Estimated runtime	In	Number		
SAP_CC_MAX_START_DELAY	Max. start delay	In	Number		
SAP_CC_CONTEXT_ID	Context ID	In	String		
SAP_CC_CONTEXT_DESCR	Context descriptions	In	String		
SAP_CC_EXE_USR_SMTP	Executioner user email address	In	String		
SAP_CC_RES_USR_SMTP	Responsible user email address	In	String		
SAP_CC_SMTP_SERVER	SMTP server	In	String		
SHOWLOG	Retrieve job log?	In	String	Y	Y, N
SHOWSPOOL	Retrieve spool list?	In	String	Y	Y, N
SHOWSTAT	Retrieve job statistics?	In	String	Y	Y, N
SHOWAPPLLOG	Retrieve application logs?	In	String	Y	Y, N
DELETE_JOB	Delete job when finished?	In	String	N	Y, N

More Information

Handling ABAP programs

SAP_ClosingCockpit_ManualTask

Use

Manual task in the context of SAP Closing Cockpit.

Parameters

Name	**Description**	**Direction**	**Data Type**
JOB_ID	Job id	In	Number
SAP_CC_CONTEXT_DESCR	Context descriptions	In	String
SAP_CC_CONTEXT_ID	Context ID	In	String
SAP_CC_ESTIMATED_RUNTIME	Estimated runtime	In	Number
SAP_CC_EXECUTIVE	Executioner	In	String

SAP_CC_EXE_USR_SMTP	Executioner user email address	In	String
SAP_CC_MAX_START_DELAY	Max. start delay	In	Number
SAP_CC_RESPONSIBLE	Responsible	In	String
SAP_CC_RES_USR_SMTP	Responsible user email address	In	String
SAP_CC_SMTP_SERVER	SMTP server	In	String
SAP_CC__GJAHR_P	Financial year (as single value)	In	String
SAP_CC__GJAHR_S	Financial year (as range)	In	String
SAP_CC__LDAY_P	Closing key day (as single value)	In	String
SAP_CC__LDAY_S	Closing key day (as range)	In	String
SAP_CC__PERIOD_P	Period (as single value)	In	String
SAP_CC__PERIOD_S	Period (as range)	In	String

SAP_CopyVariant

Use

Copy a variant from SAP CPS or from one SAP System to another.

Parameters

Name	**Description**	**Direction**	**Data Type**	**Default Expression**	**Default List of Values**
SAP_SYSTEMS	SAP System	In	String		
CLIENT	SAP client	In	String		
ABAP_PROGRAM_NAME	ABAP program	In	String		
ABAP_VARIANT_NAME	ABAP variant	In	String		
TARGET_SAP_SYSTEMS	Target SAP System	In	String		
TARGET_CLIENT	Target client	In	String		
TARGET_VARIANT	Target variant name	In	String		
WRITE_MODE	If the variant already exists in the SAP System	In	String	E (Fail)	E (Fail) S (Skip) R (Replace) C (Change)

More Information

Handling Variants

SAP_DeleteCcmsMonitors

Use

This job definition is used to delete CCMS monitors from the alert tree of SAP CPS so that these monitors are no longer fed with values from the SAP CCMS monitoring infrastructure.

Parameters

Name	Description	Data Type
MONITOR_NAME	CCMS monitor name	String
MONITOR_SET_NAME	CCMS monitor set name	String
SAP_SYSTEMS	SAP System	String

More Information

Integrating CCMS Monitors

SAP_DeleteVariant

Use

Delete a variant from an SAP System.

Parameters

Name	Description	Direction	Data Type	Default Expression
SAP_SYSTEMS	SAP System	In	String	
CLIENT	SAP client	In	String	
ABAP_PROGRAM_NAME	ABAP program	In	String	
ABAP_VARIANT_NAME	ABAP variant	In	String	
CLIENT_ONLY	Client only?	In	String	Y
DELETE_DEFINITION	Delete variant definition?	In	String	Y

More Information

Handling Variants

SAP_ExternalCommandRun

Use

This job definition is used to run external commands in SAP. This job definition requires XBP 2.0.

Parameters

Name	Description	Direction	Data Type	Default Expression
CLIENT	SAP client	In/Out	String	
DELETE_JOB	Delete job when finished?	In	String	Y
EXT_COMMAND_NAME	External command name	In	String	

EXT_COMMAND_PARAMETERS	External command parameters	In	String	
JOBCLASS	SAP job class	In	String	C
JOBCOUNT	SAP job count	Out	String	
JOBNAME	SAP job name	In	String	
LANGUAGE	SAP language for the step	In	String	
OPERATING_SYSTEM	Operating system	In	String	
SAP_SYSTEMS	SAP System	In	String	
SAP_USER_NAME	SAP user name	In	String	
SHOWLOG	Retrieve job log?	In	String	Y
STDERRCNTL	Log external errors to job log?	In	String	Y
STDOUTCNTL	Log external output to job log?	In	String	Y
TARGET_GROUP	SAP target server group name	In	String	
TARGET_HOST	Target host to run the external command	In	String	
TARGET_SERVER	SAP target server name	In	String	
TRACECNTL	Activate external trace?	In	String	N
WAIT_FOR_TERMINATION	Wait for external termination	In	String	Y

Advanced

By adding the following two parameters, you can make jobs wait on events in SAP:

Parameter	Description	Direction	Data Type
STARTCOND_EVENTID	batch event id	In	String
STARTCOND_EVENTPARM	parameter of the event id	In	String

Note that you have to duplicate the job definition and modify the duplicate.

SAP_ExternalProgramRun

Use

This job definition is used to run external programs in SAP.

Parameters

Name	Description	Direction	Data Type	Default Expression
CLIENT	SAP client	In/Out	String	
DELETE_JOB	Delete job when finished?	In	String	N
EXT_PROGRAM_NAME	External program name	In	String	
EXT_PROGRAM_PARAMETERS	External program parameters	In	String	

JOBCLASS	SAP job class	In	String	C
JOBCOUNT	SAP job count	Out	String	
JOBNAME	SAP job name	In	String	
LANGUAGE	SAP language for the step	In	String	
SAP_SYSTEMS	SAP System	In	String	
SAP_USER_NAME	SAP user name	In	String	
SHOWLOG	Retrieve job log?	In	String	Y
STDERRCNTL	Log external errors to job log?	In	String	Y
STDOUTCNTL	Log external output to job log?	In	String	Y
TARGET_GROUP	SAP target server group name	In	String	
TARGET_HOST	Target host to run the external command	In	String	
TARGET_SERVER	SAP target server name	In	String	
TRACECNTL	Activate external trace?	In	String	N
WAIT_FOR_TERMINATION	Wait for external termination	In	String	Y

Advanced

By adding the following two parameters, you can make jobs wait on events in SAP:

Parameter	Description	Direction	Data Type
STARTCOND_EVENTID	batch event id	In	String
STARTCOND_EVENTPARM	parameter of the event id	In	String

Note that you have to duplicate the job definition and modify the duplicate.

SAP_GenerateCcmsJobDefinition

Use

This job definition is used to generate a job definition in SAP CPS based on the job definition in SAP. The job name and job count must match exactly one SAP job (these parameters may **not** contain wildcard characters).

Parameters

Name	Description	Direction	Data Type	Default Expression
ADD_STEP_NUMBERS	Add step numbers to SAP job names?	In	String	N
APPLICATION	Application	In	String	
CLIENT	SAP client	In	String	
CLIENT_SPECIFIC	Generate job definitions for this client only?	In	String	N
CREATE_JOBDEFS_FOR_STEPS	Create a job definition for each step?	In	String	N

DELETE_JOB	Delete job when finished?	In	String	N
JOBCOUNT	SAP job count	In	String	
JOBDEF_NAME	Name for the generated job definition	In	String	
JOBNAME	SAP job name	In	String	
OVERWRITE	Overwrite existing job definitions?	In	String	Y
OVERWRITE_LANGUAGE	Force step language	In	String	
OVERWRITE_USER	Force step user	In	String	
SAP_SYSTEMS	SAP System	In	String	
SHOWLOG	Retrieve job log?	In	String	Y
SHOWSPOOL	Retrieve spool list?	In	String	Y

SAP_GetBatchResources

Use

This job definition is used to retrieve the batch resources of an application server of an SAP instance at a given date.

Parameters

Name	Description	Direction	Data Type
ON_DATE	On date	In	DateTimeZone
SAP_SYSTEMS	SAP System	In	String
TARGET_SERVER	SAP target server name	In	String

SAP_GetEventHistory

Use

This job definition is used to retrieve synchronized batch event history from an SAP System.

Parameters

Name	Description	Direction	Data Type
FROM_TIMESTAMP	Starting from	In	DateTimeZone
SAP_SYSTEMS	SAP System	In	String

SAP_GetInstalledComponents

Use

This job definition is used to retrieve the list of installed components of an SAP System.

Parameters

Name	Description	Direction	Data Type
SAP_SYSTEMS	SAP System	In	String

SAP_GetXmiLog

Use

Retrieves the XMI log.

Parameters

Name	Description	Direction	Data Type
SAP_SYSTEMS	SAP System	In	String
XMI_INTERFACE	XMI interface	In	String
FROM_TIMESTAMP	Start time of the XMI log	In	Date Time Zone
TO_TIMESTAMP	End time of the XMI log	In	Date Time Zone
SESSION_ID	XMI session ID	In	String
OBJECT	Object	In	String

SAP_ImportAbapParameterDefinition

Use

This job imports the parameter definition of an ABAP program.

Parameters

Name	Description	Data Type	Direction	Default Expression
SAP_SYSTEMS	SAP System	In	String	
ABAP_PROGRAM_NAME	ABAP program	In	String	
ABAP_DEF_OVERWRITE	Overwrite ABAP parameters?	In	String	Y

More Information

Importing ABAP Program Parameter Definitions

SAP_ImportAbapPrograms

Use

Import ABAP programs.

Parameters

Name	Description	Direction	Data Type	Default Expression
SAP_SYSTEMS	SAP System	In	String	
ABAP_PROGRAM_NAME	ABAP program	In	String	
MAX_COUNT	Max. count	In	Number	
DROP_OLD_SELECTION	Drop old selection?	In	String	Y

More Information

Importing ABAP Programs

SAP_ImportApplicationServers

Use

This is an internal job definition to retrieve the list of application servers in an SAP instance and is mainly used for load balancing. This job definition does not need to be submitted by a user.

Parameters

Name	Description	Direction	Data Type
SAP_SYSTEMS	SAP System	In	String

SAP_ImportArchiveObjects

Use

Imports SAP objects and the document types they support.

Note

This job definition requires XBP 3.0 enabled on the target system.

Parameters

Name	Description	Data Type	Default Expression
SAP_SYSTEMS	SAP System	In	String
DROP_OLD_SELECTION	Drop old selection?	String	Y

More Information

Importing Archive Objects

SAP_ImportBatchEvents

Use

This job definition allows you to import SAP batch events, these values are used to create drop-down menus for event history.

Note

XBP 3.0 is required.

Parameters

Name	Description	Direction	Data Type	Default Expression
SAP_SYSTEMS	SAP System	In	String	
ABAP_PROGRAM_NAME	ABAP program	In	String	
MAX_COUNT	Max. count	In	Number	
DROP_OLD_SELECTION	Drop old selection?	In	String	Y

More Information

Importing Batch Events

SAP_ImportCalendars

Use

Import SAP calendars.

Parameters

Name	Description	Direction	Data Type
SAP_SYSTEMS	SAP System	In	String
CALENDAR_TYPE	SAP Calendar type	In	String

More Information

Importing Calendars

SAP_ImportExternalCommands

Use

Import external commands.

Name	Description	Direction	Data Type	Default Expression
SAP_SYSTEMS	SAP System	In	String	

EXT_COMMAND_NAME	External command name	In	String	
OPERATING_SYSTEM	Operating system	In	String	
MAX_COUNT	Max. count	In	Number	
DROP_OLD_SELECTION	Drop old selection?	In	String	Y

More Information

Importing External Commands

SAP_ImportCcmsJobs

Use

This job definition is used to import CCMS jobs into SAP CPS. These CCMS jobs can then be scheduled to run through SAP CPS allowing you to use the advanced scheduling features of SAP CPS. The job name and job count parameters may contain wildcard characters. In this case all matching jobs will be imported. This job definition scans the SAP System for jobs matching the parameters and calls SAP_GenerateCcmsJobDefinition with the job name and job count of the matched jobs, where the actual import of the job happens. Moreover, it does conflict resolution when multiple jobs with the same name are found, it will select the latest job that matches the search criteria.

Parameters

Name	**Description**	**Direction**	**Data Type**	**Default Expression**	**Default List of Values**
SAP_SYSTEMS	SAP System	In	String		
CLIENT	SAP Client	In Out	String		
JOBNAME	SAP Job Name	In	String		
JOBCOUNT	SAP Job Count	In	String		
SAP_USER_NAME	SAP User Name	In	String		
JOB_STATUSES	Job statuses to consider	In	String		
FROM_TIMESTAMP	Starting from	In	Date Time Zone		
CLIENT_SPECIFIC	Generate Job Definitions for This Client Only?	In	String	N	Y,N
TEST_RUN	Test run?	In	String	N	Y,N
JOBDEF_NAME	Name of the Generated Job Definition	In	String		
JOBNAME_GENERATION	When multiple jobs match	In	String	P	P (Prefix job name) F (Fail)
OVERWRITE	Overwrite Existing Job Definitions?	In	String	Y	Y,N
OVERWRITE_LANGUAGE	Force Step Language	In	String		
OVERWRITE_USER	Force Step User	In	String		
APPLICATION	Application	In	String		

CREATE_JOBDEFS_FOR_STEPS	Create a Job Definition for Each Step?	In	String	N	N (Never) A (ABAP) S (Step)
EXTENDED_PRINT_PARAMETERS	Generate Extra Print Parameters?	In	String	N	Y,N
ADD_STEP_NUMBERS	Add Step Numbers to SAP Job Names?	In	String	N	Y,N
SHOWLOG	Retrieve Job Log?	In	String	Y	Y,N
SHOWSPOOL	Retrieve Spool Lists?	In	String	Y	Y,N
SHOWAPPLLOG	Retrieve Application Logs?	In	String	N	Y,N
IGNORE_APPL_RC	Ignore Application Return Code?	In	String	N	Y,N
DELETE_JOB	Delete Job When Finished?	In	String	N	Y,N

More Information

Importing SAP CCMS Jobs

SAP_ImportCcmsMonitors

Use

This job definition is used to import CCMS monitors from an SAP System. Once the monitors are imported, they will be automatically fed with values from the SAP CCMS monitoring infrastructure, provided XAL is enabled.

Parameters

Name	Description	Data Type
MONITOR_NAME	CCMS monitor name	String
MONITOR_SET_NAME	CCMS monitor set name	String
SAP_SYSTEMS	SAP System	String

SAP_ImportOutputDevices

Use

Imports output devices (printers) and the formats they support.

Note

This job definition requires XBP 2.0 enabled on the target system.

Parameters

Name	Description	Data Type	Default Expression
SAP_SYSTEMS	SAP System	String	

OUTPUT_DEVICE	The output device to import	String	*
OUTPUT_DEVICE_LONG	The long name of output device to import(XBP 3.0 only)	String	*
CHANGE_TIME	Change time stamp (XBP 3.0 only)	DateTimeZone	*
DROP_OLD_SELECTION	Drop old selection?	String	Y

More Information

Importing Output Devices

SAP_ImportRecipients

Use

This job definition is used to import the list of recipients from an SAP System, and will allow drop-down menus with all available recipients of an SAP System. This helps prevent problems due to human errors, like typos.

Parameters

Name	**Description**	**Direction**	**Data Type**	**Default Expression**
CLIENT	SAP client	In	String	
DISTRIBUTION_LIST	Distribution list	In	String	
DROP_OLD_SELECTION	Drop old matching values?	In	String	N
SAP_SYSTEMS	SAP System	In	String	
SAP_USER_NAME	SAP user name	In	String	

SAP_ImportVariantDefinition

Use

Import variants definitions of ABAP programs.

Parameters

Name	**Description**	**Direction**	**Data Type**	**Default Expression**
SAP_SYSTEMS	SAP System	In	String	
CLIENT	SAP client	In	String	
ABAP_PROGRAM_NAME	ABAP program	In	String	
ABAP_VARIANT_NAME	ABAP variant	In	String	
DROP_OLD_SELECTION	Drop old selection?	In	String	Y

More Information^M

- Handling Variants

- Importing Variants
- Importing Variant Definitions

SAP_ImportVariants

Use

Import variants of ABAP programs.

Parameters

Name	Description	Direction	Data Type	Default Expression
ABAP_PROGRAM_NAME	ABAP program	In	String	
ABAP_VARIANT_NAME	ABAP variant	In	String	
CLIENT	SAP client	In	String	
DROP_OLD_SELECTION	Drop old matching values?	In	String	Y
IMPORT_DEFINITION	Import variant definition?	In	String	
MAX_COUNT	Max. count	In	Number	
SAP_SYSTEMS	SAP System	In	String	

More Information

- Handling Variants
- Importing Variants
- Importing Variant Definitions

SAP_Info

Use

This job definition is used to retrieve information about the SAP System, like code-page, version, or list of installed components.

Parameters

Name	Description	Direction	Data Type
SAP_SYSTEMS	SAP System	In	String

SAP_InterceptedJob

Use

Note

This job definition is deprecated in favor of *SAP_CcmsJob*.

This job definition used to represent an SAP batch process that was intercepted in the SAP System. It has now been replaced with *SAP_CcmsJob* and is not supposed to be scheduled by the user.

Parameters

Name	Description	Data Type	Default Expression
CLIENT	SAP client	String	
DELETE_JOB	Delete job when finished?	String	N
JOBCOUNT	SAP job count	String	
JOBNAME	SAP job name	String	
SAP_USER_NAME	SAP user name	String	
SHOWLOG	Retrieve job log?	String	Y
SHOWSPOOL	Retrieve spool list?	String	Y

SAP_JXBP_GenerateJobDefinition

Use

This job definition is used to generate JXBP job definitions inside SAP CPS and does not to be submitted by users.

Parameters

Name	Description	Direction	Data Type	Default Expression
APPLICATION	Application	In	String	
JOBDEF_NAME	Name for the generated job definition	In	String	
JOBNAME	SAP job name	In	String	
OVERWRITE	Overwrite existing job definitions?	In	String	N
SAP_SYSTEMS	SAP System	In	String	
SYSTEM_SPECIFIC	System specific?	In	String	N
SAP_SYSTEMS	SAP System	In	String	

More Information

Importing Java Job Definitions

SAP_JXBP_ImportJobDefinitions

Use

This job definition allows you to retrieve all JXBP job definitions from the SAP Java Scheduler, these will be available in drop-down menus in SAP CPS.

Parameters

Name	Description	Direction	Data Type	Default Expression
APPLICATION	Application	In	String	
JOBDEF_NAME	Name for the generated job definition	In	String	
JOBNAME	J2EE job definition	In	String	
OVERWRITE	Overwrite existing job definitions?	In	String	N
SAP_SYSTEMS	SAP System	In	String	
SYSTEM_SPECIFIC	System specific?	In	String	N

More Information

Importing Java Job Definitions

SAP_MA_CopyParameters

Use

This job copies a mass activity parameter record and changes the parameters according to the provided parameter set.

This job definition requires the Redwood transport files to be loaded.

Parameters

Name	Description	Direction	Data Type
SAP_SYSTEMS	SAP System	In	String
CLIENT	SAP client	String	
MASS_ACTIVITY	Mass activity	In	String
MASS_ACTIVITY_PARAMETER_SET	Mass activity parameter set	In	String
MASS_ACTIVITY_SOURCE_DATE_ID	Source Date ID	In	Date Time Zone
MASS_ACTIVITY_SOURCE_ID	Source Identification	In	String
MASS_ACTIVITY_TARGET_DATE_ID	Target Date ID	In	Date Time Zone
MASS_ACTIVITY_TARGET_ID	Target Identification	In	String
MASS_ACTIVITY_DATE_SHIFT	Date shift	In	Number

SAP_MA_CreateParameters

Use

This job creates a mass activity parameter record and changes the parameters according to the provided parameter set.

This job definition requires the Redwood transport files to be loaded.

Parameters

Name	Description	Direction	Data Type
SAP_SYSTEMS	SAP System	In	String
CLIENT	SAP client	In/Out	String
MASS_ACTIVITY	Mass activity	In	String
MASS_ACTIVITY_PARAMETER_SET	Mass activity parameter set	In	String
MASS_ACTIVITY_TARGET_DATE_ID	Target Date ID	In	Date Time Zone
MASS_ACTIVITY_TARGET_ID	Target Identification	In	String

SAP_MA_DeleteDunningProposal

Use

This job deletes a dunning proposal of a new mass activity.

This job definition requires the Redwood transport files to be loaded.

Parameters

Name	Description	Direction	Data Type
SAP_SYSTEMS	SAP System	In	String
CLIENT	SAP client	In/Out	String
MASS_ACTIVITY	Mass activity	In	String
MASS_ACTIVITY_DATE_ID	Date ID	In	Date Time Zone
MASS_ACTIVITY_ID	Identification	In	String

SAP_MA_ImportMassActivities

Use

This job imports SAP mass activity definitions.

This job definition requires the Redwood transport files to be loaded.

Parameters

Name	Description	Direction	Data Type	Default Expression
SAP_SYSTEMS	SAP System	In	String	
CLIENT	SAP client	In/Out	String	
DROP_OLD_SELECTION	Drop old matching values?	In	String	N

More Information

Importing and Running Mass Activities

SAP_MA_ImportMassActivity

Use

This job imports an SAP mass activity definition.

This job definition requires the Redwood transport files to be loaded.

Parameters

Name	Description	Direction	Data Type	Default Expression
SAP_SYSTEMS	SAP System	In	String	
CLIENT	SAP client	In/Out	String	
MASS_ACTIVITY	Mass activity	In	String	
MASS_ACTIVITY_TCODE	Mass activity transaction code	In	String	
DROP_OLD_SELECTION	Drop old matching values?	In	String	N

More Information

Importing and Running Mass Activities

SAP_MA_MassActivityRun

Use

This job creates a parameter record and starts a mass activity run.

This job definition requires the Redwood transport files to be loaded.

Parameters

Name	Description	Direction	Data Type
SAP_SYSTEMS	SAP System	In	String
CLIENT	SAP client	In/Out	String
MASS_ACTIVITY	Mass activity	In	String
MASS_ACTIVITY_PARAMETER_SET	Mass activity parameter set	In	String
MASS_ACTIVITY_DATE_ID	Date ID	In/Out	Date Time Zone
MASS_ACTIVITY_ID	Identification	In/Out	String

More Information

Importing and Running Mass Activities

SAP_MA_MassActivityTemplateRun

Use

This job starts a mass activity run using an existing parameter record.

This job definition requires the Redwood transport files to be loaded.

Parameters

Name	Description	Direction	Data Type
SAP_SYSTEMS	SAP System	In	String
CLIENT	SAP client	In/Out	String
MASS_ACTIVITY	Mass activity	In	String
MASS_ACTIVITY_DATE_ID	Date ID	In	Date Time Zone
MASS_ACTIVITY_ID	Identification	In	String

More Information

Importing and Running Mass Activities

SAP_MA_ResumeMassActivity

Use

This job resumes processing of an interrupted SAP mass activity run.

This job definition requires the Redwood transport files to be loaded.

Parameters

Name	Description	Direction	Data Type
SAP_SYSTEMS	SAP System	In	String
CLIENT	SAP client	In/Out	String
MASS_ACTIVITY	Mass activity	In	String
MASS_ACTIVITY_DATE_ID	Date ID	In	Date Time Zone
MASS_ACTIVITY_ID	Identification	In	String

More Information

Importing and Running Mass Activities

SAP_MA_ShowMassActivity

Use

This job shows the definition of an SAP mass activity.

This job definition requires the Redwood transport files to be loaded.

Parameters

Name	Description	Direction	Data Type
SAP_SYSTEMS	SAP System	In	String
CLIENT	SAP client	In/Out	String
MASS_ACTIVITY	Mass activity	In	String

SAP_MA_StopMassActivity

Use

This job stops an SAP mass activity run.

This job definition requires the Redwood transport files to be loaded.

Parameters

Name	Description	Direction	Data Type
SAP_SYSTEMS	SAP System	In	String
CLIENT	SAP client	In/Out	String
MASS_ACTIVITY	Mass activity	In	String
MASS_ACTIVITY_DATE_ID	Date ID	In	Date Time Zone
MASS_ACTIVITY_ID	Identification	In	String

SAP_MaintainXbpOptions

Use

This job maintains XBP options for interception and parent/child functionality.

Parameters

Name	Description	Direction	Data Type	Default Expression	Default List of Values
SAP_SYSTEMS	SAP System	In	String		
INTERCEPTION_ACTION	Interception action	In	String	R (Show)	R (Show) S (Enable) 3 (Enable with XBP 3.0 features) C (Disable)
PARENTCHILD_ACTION	Parent/Child action	In	String	R (Show)	R (Show) S (Enable) C (Disable)
INTERCEPTION_OPTION	Interception enabled?	Out	String		
PARENTCHILD_OPTION	Parent/Child enabled?	Out	String		

Values

The job will output the current setting when run with R (Show), change the setting and output the result when you use the other values.

Code	Description
S	Enabled
C	Disabled
3	XBP 3.0 features enabled (valid only for interception)

The output can be found in the out parameters on the job and in the output file. Note that to use the XBP 3.0 features your system needs to have XBP 3.0.

More Information

- Understanding the Different XBP Interface Releases
- Enable XBP 2.0 or 3.0 Functionality
- Enhanced SAP Interfaces with Redwood Transports

SAP_PI_GetChannelStatus

Purpose

Retrieves the current status of an SAP XI/PI channel.

Parameters

Parameter	Description	Direction	Data Type
SAP_SYSTEMS	SAP system	In	String
CHANNEL_NAME	Channel name	In	String
CHANNEL_ID	Channel id	In	String
CHANNEL_STATUS	Channel activation status	Out	String
CHANNEL_STATE	Channel state	Out	String

SAP_PI_ImportChannels

Purpose

Imports SAP XI/PI channels so you can control the channels from within SAP CPS

Parameters

Parameter	Description	Direction	Data Type	Default Value
SAP_SYSTEMS	SAP system	In	String	
CHANNEL_NAME	Channel name	In	String	

CHANNEL_SERVICE	Channel service	In	String	
CHANNEL_PARTY	Channel party	In	String	
DROP_OLD_SELECTION	Drop old matching values?	In	String	N

More Information

Importing Communication Channels

SAP_PI_StartChannel

Purpose

Starts an SAP XI/PI channel

Parameter

Parameter	**Description**	**Direction**	**Data Type**
SAP_SYSTEMS	SAP system	In	String
CHANNEL_NAME	Channel name	In	String
CHANNEL_ID	Channel id	In	String
CHANNEL_STATUS	Channel activation status	Out	String
CHANNEL_STATE	Channel state	Out	String

SAP_PI_StopChannel

Purpose

Stops an SAP XI/PI channel

Parameter

Parameter	**Description**	**Direction**	**Data Type**
SAP_SYSTEMS	SAP system	In	String
CHANNEL_NAME	Channel name	In	String
CHANNEL_ID	Channel id	In	String
CHANNEL_STATUS	Channel activation status	Out	String
CHANNEL_STATE	Channel state	Out	String

SAP_RaiseEvent

Use

This job definition is used to raise a batch event in SAP.

Parameters

Name	Description	Data Type
SAP_SYSTEMS	SAP System	String
EVENT_ID	Event ID	String
EVENT_PARAMETER	Event parameter	String

SAP_ReloadApplicationLinks

Use

This job definition reloads the SAP Application links.

Parameters

Name	Description	Direction	Data Type	Default Expression
DROP_OLD_SELECTION	Drop old matching values?	In	String	N
SAP_SYSTEMS	SAP System	In	String	

SAP_ReleaseInterceptedJobs

Use

Releases intercepted SAP jobs.

Parameters

Name	Description	Direction	Data Type
CLIENT	SAP client	In	String
JOBNAME	SAP job name	In	String
SAP_SYSTEMS	SAP System	In	String
SAP_USER_NAME	SAP user name	In	String

SAP_RunTemplateJob

Use

This job definition is used to run a job that exists in SAP by making a copy of this job and running this copy. The original job is not changed. Only jobs with ABAP steps can be run by this job definition.

Note

Jobs with steps containing external programs or commands require Chargeable version of SAP CPS

Parameters

Name	Description	Data Type	Default Expression
DELETE_JOB	Delete job when finished?	String	N
FROM_STEP	From step (0, 1 = all steps)	Number	
JOBCOUNT	SAP job count	String	
JOBNAME	SAP job name	String	
SAP_USER_NAME	SAP user name	String	
SHOWLOG	Retrieve job log?	String	Y
SHOWSPOOL	Retrieve spool list?	String	Y
TARGET_GROUP	SAP target group name	String	
TARGET_JOBNAME	SAP target job name	String	
TARGET_SERVER	SAP target application server name	String	

SAP_SetXmiAuditLevel

Use

This job definition lets you manage the audit level of an XMI interface. Valid audit levels are integers ranging from `0` to `3`.

Parameters

Name	Description	Data Type
SAP_SYSTEMS	SAP System	String
AUDIT_LEVEL	Audit level	Number
XMI_INTERFACE	XMI interface	String

SAP_SolutionManager_AbapRun

Use

Run an ABAP report from SAP Solution Manager.

Parameters

Name	Description	Direction	Data Type	Default Expression	Default List of Values
SAP_SYSTEMS	SAP System	In	String		
CLIENT	SAP client	In	String		
JOBNAME	SAP job name	In	String		
ABAP_PROGRAM_NAME	ABAP program	In	String		

ABAP_VARIANT_NAME	ABAP variant	In	String		
TEMPORARY_VARIANT	Create temporary variant?	String	N	Y,N	
JOBCLASS	SAP job class	In	String	C	A, B, C
SAP_USER_NAME	SAP user name	In	String		
TARGET_SERVER	SAP target application server name	In	String		
JOBDOCID	Job documentation ID	In	String		
PRINT_ARMOD	Print/Archive mode	In	String	1 (Print only)	1 (Print only) 2 (Archive only) 3 (Print and Archive)
PRINT_PDEST	Printer	In	String		
PRINT_PRTXT	Cover sheet title	In	String		
PRINT_PRCOP	# of copies	In	String	1	
PRINT_PRNEW	New spool?	In	String	Y	Y, N
PRINT_PLIST	Spool list name	In	String		
PRINT_PRIMM	Print immediately?	In	String	N	Y, N
PRINT_PRREL	Release print job?	In	String	N	Y, N
PRINT_PEXPI	Expiration in days	In	String	8	0 (Do not Delete)1-8 days
PRINT_LICT	Print lines per page	In	String		
PRINT_LISZ	Print columns per page	In	String		
PRINT_PAART	Print format	In	String		
PRINT_PRBIG	Selection cover sheet?	In	String	N	Y, N
PRINT_PRSAP	SAP cover page?	In	String	D (System Default)	D (System Default)Y (Print)N (Do not print)
PRINT_PRREC	Spool list recipient	In	String		
PRINT_PRABT	Department on cover page	In	String		
PRINT_PRBER	Authority for printing	In	String		
PRINT_PRDSN	Spool dataset name	In	String		
PRINT_PTYPE	Spool type	In	String		
PRINT_FOOTL	Print a footer?	In	String	N	Y, N
PRINT_PRUNX	OS cover sheet?	In	String	D (System Default)	D (System Default)Y (Print)N (Do not print)
ARCHIVE_SAP_OBJECT	Archive SAP object	In	String		
ARCHIVE_AR_OBJECT	Document type	In	String		
ARCHIVE_INFO	Archive info	In	String		
ARCHIVE_ARCTEXT	Archive text information field	In	String		

REC_RECIPIENT	Spool list recipient	In	String		
REC_TYPE	Recipient type	In	String	U (Internet User)	U (Internet User) B (SAP user) O (SAP-Office user) P (Personnel distribution list) C (Shared distribution list) F (Fax number)
REC_COPY	Recipient copy flag	In	String	N	Y, N
REC_BLIND_COPY	Recipient blind copy flag	In	String	N	Y, N
REC_EXPRESS	Recipient express flag	In	String	N	Y, N
REC_NO_FORWARDING	Recipient no forwarding flag	In	String	N	Y, N
REC_NO_PRINT	Recipient no print flag	In	String	N	Y, N
REC_DELIVER	Report send status	In	String	_ (System Default)	_ (System Default) A (Always) E (On error) N (Never)
REC_MAILSTATUS	Report status by e-mail	In	String	_ (System Default)	_ (System Default) A (Always) E (On error) N (Never)
SHOWLOG	Retrieve job log?	In	String	Y	Y, N
SHOWSPOOL	Retrieve spool list?	In	String	Y	Y, N
SHOWSTAT	Retrieve job statistics?	In	String	Y	Y, N
SHOWAPPLLOG	Retrieve application logs?	In	String	Y	Y, N
DELETE_JOB	Delete job when finished?	In	String	N	Y, N

More Information

Handling ABAP programs

SAP_StoreVariant

Use

Store a variant of an ABAP program.

Parameters

Parameter	**Description**	**Data Type**	**Default Value**	**Default List of Values**
SAP_SYSTEMS	The SAP System	String		
CLIENT	The client of the SAP System	String		
ABAP_PROGRAM_NAME	The variants ABAP program	String		
ABAP_VARIANT_NAME	Variant	String		

WRITE_MODE	The write mode allows you to specify how the job will handle duplicate entries.	String	E (Error) - the job gets the status error, nothing is changed in the SAP System	E (Error) - the job gets the status error, nothing is changed in the SAP System S (Skip) - the variant is skipped R (Replace) - the variant is overwritten with the variant definition from SAP CPS C (Change) - only different parameters are changed, deleted variant parameters will not be removed (recommended)

SAP_SynchronizeEventHistoryCriteria

Use

This job definition is used to synchronize batch event criteria in SAP. This job definition requires XBP 3.0. The *Additional Conditions* parameter allows you to use BNF expressions to create complex expressions, the syntax is as follows:

```
<expr>   ::= <term> { OR <term> }
<term>   ::= <factor> { AND <factor> }
<factor> ::= [ NOT ] <mask>
<mask>   ::= event id mask
```

Example

In the following example, you include all events in event history which have names (event id's) starting with `EventBI` or `EventBW`; you exclude the `EventBITest` event and all events starting with `EventBWtest`.

```
 not EventBITest and EventBI* or EventBW* and not EventBWtest*
```

Parameters

Name	Description	Direction	Data Type	Default Expression	Default List of Values
ACTIVATE_RULES	Activate rules after synchronization	In	String	Y	
ADDITIONAL_CONDITIONS	Additional Conditions	In	String		
SAP_SYSTEMS	SAP System	In	String		

SAP_SynchronizeInterceptionCriteria

Use

This job definition synchronizes interception criteria (job control rules) between SAP CPS and SAP. This job definition requires XBP 2.0.

Parameters

Name	Description	Data Type	Default Expression	Default List of Values
OVERWRITE	Append or overwrite criteria in SAP?	String	A	A (Append) O (Overwrite)
SAP_SYSTEMS	SAP System	String		
SYNC_DIRECTION	Synchronization direction	String	N	N (Show Only) S (SAP -> Scheduler) B (SAP <-> Scheduler) C (Scheduler -> SAP)

SYNC_DIRECTION

- Show Only : view option, display current interception criteria as defined in the SAP System as

well as job control rules of SAP CPS.

- Scheduler -> SAP: all SAP CPS rules are transferred to SAP

and afterwards the SAP table contains the same rules as in SAP CPS.

- With Scheduler <-> SAP: all roles both from SAP CPS and SAP are

synchronized. That means the rules of SAP are going into the SAP CPS tables

and the rules of SAP CPS are going to the SAP tables. Final result is that

both tables are equal with respect to the rule-definitions.

- With SAP -> Scheduler: all SAP rules are transferred to SAP CPS

and afterwards the SAP CPS table contain the same rules as in SAP.

OVERWRITE

Append and Overwrite options only work for syn_direction SAP CPS -> SAP and

SAP CPS <-> SAP.

- Append: keeps the interception criteria that are already in

the SAP table intact and adds our rules.

- Overwrite: replaces all interception

criteria that might be defined in SAP by our rules.

SAP_SynchronizeTerminatedCcmsJobs

Use

This job definition synchronizes job states between SAP CPS and SAP. Normally, customers will schedule this job definition to run regularly, for example once per day.

Parameters

Name	Description	Direction	Data Type	Default Expression
SAP_SYSTEMS	SAP System	In	String	
DELETE_JOBS_OLDER_THEN	Delete terminated jobs that are older then that number of days	Number		
MAX_ALLOWED_EXCEPTIONS	Max consecutive exceptions allowed	Number		

Process Server Parameters

Scripting Contexts and Implicit Objects

Use

Scripting contexts are fields in which scripting, with Redwood Expression Language or RedwoodScript may be used. The implicit objects provided differ between the contexts.

This section lists the implicit objects by category, and what they contain.

Implicit object	Description	Class
parameters.*name*	values of parameters of current job	java.lang.String, java.lang.Number, java.util.Date, com.redwood.scheduler.api.date.DateTimeZone
outParameters.*name*	values of Out parameters of current job	java.lang.String, java.lang.Number, java.util.Date, com.redwood.scheduler.api.date.DateTimeZone
chainParameters.*name*	values of parameters of inner-most job chain	java.lang.String, java.lang.Number, java.util.Date, com.redwood.scheduler.api.date.DateTimeZone
chainOutParameters.*name*	values of out parameters of inner-most job chain	java.lang.String, java.lang.Number, java.util.Date, com.redwood.scheduler.api.date.DateTimeZone
JobChainParameters.getOutValueString(<jobName>, <parameter>)	The value of the out parameter of the job in chain (see placeholders below)	java.util.String
JobChainParameters.getOutValueNumber(jobName>, <parameter>)	The value of the out parameter of the job in chain (see placeholders below)	java.lang.Number
JobChainParameters.getOutValueDate(jobName>, <parameter>)	The value of the out parameter of the job in chain (see placeholders below)	java.util.Date
chainJobId	job id of the inner-most chain (needs to be a runtime parameter)	long
jobId	job id of the current job (needs to be a runtime parameter)	long

topLevelJobId	job id of highest parent job in hierarchy (needs to be a runtime parameter)	long
returnCode	return code of current job	long
stepIterations	iteration number of the current step or job	long
waitEvents.*name*.finalPath	the path of the file that raised the event (after move, if applicable)	java.lang.String
waitEvents.*name*.originalPath	the path of the file that raised the event (before move, if applicable)	java.lang.String
waitEvents.*name*.raiserComment	the comment of the raiser	java.lang.String

The placeholders for the *JobChainParameters.getOutValue* functions are as follows:

- <jobName> - the name of the job, for example:
  - `Step 1, job 2`
  - `Extract data, job 4`
- <parameter> - the name of the parameter

Example

```
=JobChainParameters.getOutValueString('Step 1, Job 1', 'Param1')
```

Scripting Contexts

Contexts are places in the interface where Redwood Expression Language can be used. The variables added with the *Scripting* module are only available in RedwoodScript, not in Redwood Expression Language.

Please check your license to make sure you have the Scripting Module.

They are additional to those available in the Free of Charge version.

All contexts have the following imports:

- com.redwood.scheduler.api.model.enumeration - This defines all the enumerations used by the API
- com.redwood.scheduler.infrastructure.logging.enumeration- This defines the Level enumeration

Job Definition Parameter Default Expressions

Redwood Expression Language

- parameters
- waitEvents

Job Definition Parameter Constraints

Redwood Expression Language

- Constraint

- Table

Job Chain Step Preconditions

Redwood Expression Language

- chainParameters
- chainOutParameters
- stepIterations
    - The amount of times this job has been called in this instance of the job chain.
- JobChainParameters.getOutValueString(jobName, parameter)
- JobChainParameters.getOutValueNumber(jobName, parameter)
- JobChainParameters.getOutValueDate(jobName, parameter)

Job Chain Job Preconditions

Redwood Expression Language

- parameters
- outParameters
- chainParameters
- chainOutParameters
- stepIterations
    - The amount of times this job has been called in this instance of the job chain.
- JobChainParameters.getOutValueString(jobName, parameter)
- JobChainParameters.getOutValueNumber(jobName, parameter)
- JobChainParameters.getOutValueDate(jobName, parameter)

Job Chain Job Parameter Mappings

Redwood Expression Language

- parameters
- outParameters
- chainParameters
- chainOutParameters
- chainJobId
- topLevelJobId

- JobChainParameters.getOutValueString(jobName, parameter)
- JobChainParameters.getOutValueNumber(jobName, parameter)
- JobChainParameters.getOutValueDate(jobName, parameter)

Job Chain Job Scheduling Parameters

Redwood Expression Language

- parameters
- outParameters
- chainParameters
- chainOutParameters
- JobChainParameters.getOutValueString(jobName, parameter)
- JobChainParameters.getOutValueNumber(jobName, parameter)
- JobChainParameters.getOutValueDate(jobName, parameter)

RedwoodScript Job Definitions

Redwood Expression Language

- Feature not available

With *Scripting* module:

RedwoodScript

Object	**Class**	**Example Code**
jcsOutLog	com.redwood.scheduler.infrastructure.logging.api.Logger	jcsOutLog.warning("This is a warning!");
jcsErr	java.io.PrintWriter	jcsErr.println("This line gets printed to stderr of the job.");
jcsJobContext	com.redwood.scheduler.api.job.UserJobContext	jcsJobContext.waitForAllChildren();
jcsSession	com.redwood.scheduler.api.model.SchedulerSession	String strUser = jcsSession.getUserName();
jcsErrLog	com.redwood.scheduler.infrastructure.logging.api.Logger	jcsErrLog.warning("This is a warning!");
jcsJob	com.redwood.scheduler.api.model.Job	jcsJob.restart();
jcsOut	java.io.PrintWriter	jcsOut.println("This line gets printed to stdout of the job.");
<parameter_name>	java.lang.String, java.lang.Number, java.util.Date, com.redwood.scheduler.api.date.DateTimeZone	

RedwoodScript in the Shell

Redwood Expression Language

- Feature not available

With *Scripting* module:

RedwoodScript

Object	**Class**	**Example Code**
jcsSession	com.redwood.scheduler.api.model.SchedulerSession	String strUser = jcsSession.getUserName();
jcsOut	java.io.PrintWriter	jcsOut.println("This text will be printed to the standard output.");
jcsErr	java.io.PrintWriter	jcsOut.println("This text will be printed to the standard error.");

Job definition On-Change Actions

Redwood Expression Language

- Feature not available

With *Scripting* module:

RedwoodScript

Object	**Class**	**Example Code**
jcsOutLog	com.redwood.scheduler.infrastructure.logging.api.Logger	jcsOutLog.warning("This is a warning!");
jcsOnChangeContext	com.redwood.scheduler.api.scripting.variables.PreExecutingActionScriptObject	jcsOnChangeContext.setFailJobOnError(false);
jcsSession	com.redwood.scheduler.api.model.SchedulerSession	String strUser = jcsSession.getUserName();
jcsJob	com.redwood.scheduler.api.model.Job	jcsJob.hold();
jcsOut	java.io.PrintWriter	jcsOut.println("This text will be printed to a specific output file on the job.");

Job Definition Pre-Running Actions

Redwood Expression Language

- Feature not available

With *Scripting* module:

RedwoodScript

Object	**Class**	**Example Code**
jcsOutLog	com.redwood.scheduler.infrastructure.logging.api.Logger	jcsOutLog.info("This is an informational message!")

jcsSession	com.redwood.scheduler.api.model.SchedulerSession	String strUser = jcsSession.getUserName();
jcsJob	com.redwood.scheduler.api.model.Job	jcsJob.hold();
jcsPreRunningContext	com.redwood.scheduler.api.scripting.variables.PreExecutingActionScriptObject	
jcsOut	java.io.PrintWriter	jcsOut.println("This text will be printed to a specific output file on the job.");

Job Definition Post-Running Actions

Redwood Expression Language

- Feature not available

With *Scripting* module:

RedwoodScript

Object	**Class**	**Example Code**
jcsOutLog	com.redwood.scheduler.infrastructure.logging.api.Logger	jcsOutLog.debug("This is a debug message!")
jcsPostRunningContext	com.redwood.scheduler.api.scripting.variables.PostRunningActionScriptObject	jcsPostRunningContext.setFailJobOnError(false);
jcsSession	com.redwood.scheduler.api.model.SchedulerSession	String strUser = jcsSession.getUserName();
jcsJob	com.redwood.scheduler.api.model.Job	jcsJob.restart();
jcsOut	java.io.PrintWriter	jcsOut.println("This text will be printed to a specific output file on the job.");

Trigger Before Definition Change

Redwood Expression Language

- Feature not available

With *Scripting* module:

RedwoodScript

Object	**Class**	**Example Code**
jcsOutLog	com.redwood.scheduler.infrastructure.logging.api.Logger	jcsOutLog.fatal("This is a fatal error message!");
jcsTriggerContext	com.redwood.scheduler.api.scripting.variables.TriggerScriptObject	jcsTriggerContext.getSchedulerEntity();

Trigger Before Job On Change

Redwood Expression Language

- Feature not available

With *Scripting* module:

RedwoodScript

Object	**Class**	**Example Code**
jcsOutLog	com.redwood.scheduler.infrastructure.logging.api.Logger	jcsOutLog.warning("This is a warning!");
jcsOnChangeContext	com.redwood.scheduler.api.scripting.variables.PreExecutingActionScriptObject	jcsOnChangeContext.setFailJobOnError(false);
jcsSession	com.redwood.scheduler.api.model.SchedulerSession	String strUser = jcsSession.getUserName();
jcsJob	com.redwood.scheduler.api.model.Job	jcsJob.hold();
jcsOut	java.io.PrintWriter	jcsOut.println("This text will be printed to a specific output file on the job.");

Trigger Before Job Pre Running

Redwood Expression Language

- Feature not available

With *Scripting* module:

RedwoodScript

Object	**Class**	**Example Code**
jcsOutLog	com.redwood.scheduler.infrastructure.logging.api.Logger	jcsOutLog.info("This is an informational message!")
jcsSession	com.redwood.scheduler.api.model.SchedulerSession	Queue jcsJobQueue = jcsSession.getQueueByName("UNIX_Generic");
jcsJob	com.redwood.scheduler.api.model.Job	jcsJob.hold();
jcsPreRunningContext	com.redwood.scheduler.api.scripting.variables.PreExecutingActionScriptObject	
jcsOut	java.io.PrintWriter	jcsOut.println("This text will be printed to a specific output file on the job.");

Trigger Before Job Post Running

Redwood Expression Language

- Feature not available

With *Scripting* module:

RedwoodScript

Object	**Class**	**Example Code**
jcsOutLog	com.redwood.scheduler.infrastructure.logging.api.Logger	jcsOutLog.debug("This is a debug message!")
jcsPostRunningContext	com.redwood.scheduler.api.scripting.variables.PostRunningActionScriptObject	

jcsSession	com.redwood.scheduler.api.model.SchedulerSession	String strUser = jcsSession.getUserName();
jcsJob	com.redwood.scheduler.api.model.Job	jcsJob.restart();
jcsOut	java.io.PrintWriter	jcsOut.println("This text will be printed to a specific output file on the job.");

Import Rule Sets

Redwood Expression Language

- Feature not available

With *Scripting* and *Import Export* modules:

RedwoodScript

Object	**Class**	**Example Code**
jcsSession	com.redwood.scheduler.api.model.SchedulerSession	Queue jcsJobQueue = jcsSession.getQueueByName("PRD_QUEUE");
jcsImportRuleSet	com.redwood.scheduler.api.scripting.variables.ImportActionScriptObject	jcsImportRuleSet.getObjects();

Alerting Source Actions

Redwood Expression Language

- Feature not available

With *Scripting* and *Alerting* modules:

RedwoodScript

Object	**Class**	**Example Code**
jcsOutLog	com.redwood.scheduler.infrastructure.logging.api.Logger	jcsOutLog.info("This is an informational message");
jcsAlertSourcePostAlertContext	com.redwood.scheduler.api.scripting.variables.AlertSourcePostAlertActionScriptObject	AlertSource jcsAlertInitiator = jcsAlertSourcePostAlertContext.getAlertSource();

Alert Escalation Actions

Redwood Expression Language

- Feature not available

With *Scripting* and *Alerting* modules:

RedwoodScript

Object	**Class**	**Example Code**
jcsOutLog	com.redwood.scheduler.infrastructure.logging.api.Logger	jcsOutLog.info("This is an informational message");

jcsAlertEscalationPostAlertContext	com.redwood.scheduler.api.scripting.variables.AlertEscalationPostAlertActionScriptObject	jcsAlertEscalationPostAlertContext.isResend()

Alert Email Gateway Actions

Redwood Expression Language

- Feature not available

With *Scripting* and *Alerting* modules:

RedwoodScript

Object	**Class**	**Example Code**
jcsOutLog	com.redwood.scheduler.infrastructure.logging.api.Logger	jcsOutLog.info("This is an informational message");
jcsEmailAlertGatewayPreSendActionContext	com.redwood.scheduler.api.scripting.variables.EmailAlertGatewayPreSendActionScriptObject	jcsEmailAlertGatewayPreSendActionContext.getMessage();

Using RedwoodScript in Objects

Object	**Topic**
Jobs	RedwoodScript in Jobs
Queues	RedwoodScript in Queues
Process Servers	RedwoodScript in Process Servers

REL Functions

Built in function classes

- BaseCast - Prefix: None.
- BaseString - Prefix: None.
- BaseCast - Prefix: Casts
- BaseMath - Prefix: Math
- BaseRange - Prefix: Range
- BaseTime - Prefix: Time
- ExtendedRange - Prefix: Range
- ExtendedTime - Prefix: Time
- ModuleScriptingConstraint - Prefix: Constraint
- ModuleScriptingJobChainParameters - Prefix: JobChainParameters
- ModuleScriptingQuery - Prefix: Query

- ModuleScriptingSap - Prefix: SAP
- ModuleScriptingTable - Prefix: Table
- ModuleScriptingVariable - Prefix: Variable

Note

The *current<data_type>* object designates the object you want to use with the function.

Built in functions

Cast

Prefix:

Availability: Always.

Change a data type to another.

ToBoolean

Signature: `Object ToBoolean(Object arg)`

Availability: Always.

Convert *arg* to a boolean (true/false) if possible.

ToNumber

Signature: `Object ToNumber(Object arg)`

Availability: Always.

Convert *arg* to a number if possible.

ToInteger

Signature: `Object ToInteger(Object arg)`

Availability: Always.

Convert *arg* to an integer if possible.

ToString

Signature: `Object ToString(Object arg)`

Availability: Always.

Convert *arg* to a string.

ToYN

Signature: `Object ToYN(Object arg)`

Availability: Always.

Convert *arg* to Y for true and N for any other value.

Math

Prefix: Math

Availability: Always.

Mathematical functions for use with *Numbers*.

abs

Signature: `BigDecimal Math.abs(BigDecimal x)`

Availability: Always.

Calculate the absolute value of *x*.

floor

Signature: `BigDecimal Math.floor(BigDecimal x)`

Availability: Always.

Calculate the floor (greatest integer less than) of *x*.

ceil

Signature: `BigDecimal Math.ceil(BigDecimal x)`

Availability: Always.

Calculate the ceiling (least integer greater than) of *x*.

round

Signature: `BigDecimal Math.round(BigDecimal x)`

Availability: Always.

Round *x*. If *x* ends in .5, it will be rounded up.

Range

Prefix: Range

Availability: Always.

Functions for working with sets and ranges. Set is written as "XXXXXXX" where X is either '_' or a letter and the first X represents 0 (or Sunday for days of the week, or January for months). Any letter is in the set, and '_' is not in the set. Example sets: Ranges are made up of comma separated sections, each of which is a number (eg. 1) or a range of numbers (eg. 4-7). All numbers specified are in the range, other numbers are not.

Set	**X's**	**English**
First month of the quarter	`X__X__X__X__`	`J__A__M__O__`
Last month of the quarter	`__X__X__X__X`	`__M__J__S__D`

Not last month of the quarter	`XX_XX_XX_XX_`	`JF_AM_JA_ON_`
Monday, Wednesday, Friday	`_X_X_X_`	`_M_W_F_`
Saturday, Sunday	`X_____X`	`S_____S`

inRange

Signature: `boolean Range.inRange(int candidate, String trueRange)`

Availability: Always.

Return true if *candidate* is in the range specified by *trueRange*.

inSet

Signature: `boolean Range.inSet(int candidate, String trueSet)`

Availability: Always.

Return true if *candidate* is in the set specified by *trueRange*.

String

Prefix:

Availability: Always.

Functions for working with *Strings*.

currentObject.concat

Signature: `String currentObject.concat(String s1, ...)`

Availability: Always.

Concatenate strings to *currentObject*. Any number of other objects may be passed as arguments.

currentObject.toString

Signature: `String currentObject.toString()`

Availability: Always.

Convert *currentObject* to a string.

currentString.charAt

Signature: `String currentString.charAt(int pos)`

Availability: Always.

Get character at *pos* in *currentString*.

currentString.indexOf

Signature: `int currentString.indexOf(String searchFor, [int startIndex])`

Availability: Always.

Get the first index of the string *searchFor* in *currentString*, optionally starting at startIndex (default=0).

currentString.lastIndexOf

Signature: `int currentString.lastIndexOf(String searchFor, [int startIndex])`

Availability: Always.

Get the last index of the string *searchFor* in the *currentString*, optionally starting at startIndex (default=0).

currentString.split

Signature: `String[] currentString.split(String separator, int limit)`

Availability: Always.

Split *currentString* into an array using the specified *separator*. Limit the length of the array to *limit* elements.

currentString.substring

Signature: `String currentString.substring([int startIndex] [, int endIndex])`

Availability: Always.

Get a substring of *currentString*, starting at *startIndex* (defaults to zero) and ending at *endIndex* (defaults to the end of the string).

currentString.toLowerCase

Signature: `String currentString.toLowerCase()`

Availability: Always.

Convert *currentString* to lower case.

currentString.toUpperCase

Signature: `String currentString.toUpperCase()`

Availability: Always.

Convert *currentString* to upper case

Time

Prefix: Time

Availability: Always.

Time expressions are a sequence of operations, applied in order to a time (generally now, but may be a specified date). They may be: The <value> is always a number: The <specifier> is one of: Plurals with an additional 's' are accepted (eg. days, weeks, ...). The day is always the day of the month, and dow is the day of the week. Example time expressions:

- set <specifier> <value> - set <specifier> to <value>

- add <value> <specifier> - add <value> to <specifier> (This may propagate)
- subtract <value> <specifier> - subtract <value> from <specifier> (This may propagate)
- truncate <specifier> - truncate at <specifier> (This will zero everything below <specifier>).
- The days of the week start at 1 for Sunday.
- The days of the year start at 1 for 1 January.
- add/subtract: second, minute, hour, day, week, month, year
- truncate: second, minute, hour, day, month
- set: second, minute, hour, day, dow, day_of_year, week, week_of_month, month
- add 1 minute
- add 3 seconds
- set hour 1
- truncate day
- subtract 2 days

now

Signature: `DateTimeZone Time.now([String timeZoneName])`

Availability: Always.

Return the time in the context time zone (generally the time zone of the job). This can be optionally overridden by specifying the Olson name of a time zone as *timeZoneName*.

expression

Signature: `DateTimeZone Time.expression(Object date, String expression)`

Availability: Always.

Apply *expression* to the *date* specified.

expressionNow

Signature: `DateTimeZone Time.expressionNow(String expression)`

Availability: Always.

Apply *expression* to the current time.

isTimeWindowOpenNow

Signature: `boolean Time.isTimeWindowOpenNow(String timeWindow)`

Availability: Always.

Return true if the time window *timeWindow* is open now, false otherwise.

isTimeWindowOpen

Signature: `boolean Time.isTimeWindowOpen(Object date, String timeWindow)`

Availability: Always.

Return true if the time window *timeWindow* is open on the specified *date*, false otherwise.

Range

Prefix: Range

Availability: RedwoodScript only. Requires 'Module.AdvancedScripting' key.

Extended *Range* function with notification.

inRangeWarning

Signature: `boolean Range.inRangeWarning(int candidate, String trueRange, String warningRange, int severity, String message)`

Availability: RedwoodScript only. Requires 'Module.AdvancedScripting' key.

Return true if *candidate* is in the range specified by *trueRange*. Return true, but log a warning operator message if *candidate* is in *warningRange*. The operator message is specified by the *message* and *severity* parameters.

Prefix:

Availability: RedwoodScript only. Requires 'Module.AdvancedScripting' key.

Functions for accessing the data model.

getParentJobId

Signature: `String .getParentJobId(String jobId)`

Availability: RedwoodScript only. Requires 'Module.AdvancedScripting' key.

Get the parent job id of the specified job id, or null if the job id is null or its parent is null.

queryHTMLTable

Signature: `int .queryHTMLTable(String query [, Object [] bindVariables])`

Availability: RedwoodScript only. Requires 'Module.AdvancedScripting' key.

Produce HTML output on stdout for the query *query* with optional bind variables bindVariables. Bind variable place holders are specified as ? in the query.

queryCSV

Signature: `int .queryCSV(String query [, Object [] bindVariables])`

Availability: RedwoodScript only. Requires 'Module.AdvancedScripting' key.

Produce CSV output on stdout for the query *query* with optional bind variables bindVariables. Bind variable place holders are specified as ? in the query.

exists

Signature: `boolean .exists(VisitorContext context, String statement, Object [] bindVariables)`

Availability: RedwoodScript only. Requires 'Module.AdvancedScripting' key.

queryDateTimeZone

Signature: `DateTimeZone .queryDateTimeZone(VisitorContext context, String statement, Object [] bindVariables)`

Availability: RedwoodScript only. Requires 'Module.AdvancedScripting' key.

queryDate

Signature: `Date .queryDate(VisitorContext context, String statement, Object [] bindVariables)`

Availability: RedwoodScript only. Requires 'Module.AdvancedScripting' key.

Time

Prefix: Time

Availability: RedwoodScript only. Requires 'Module.AdvancedScripting' key.

Advanced time functions.

isDayOfWeekInSetNow

Signature: `boolean Time.isDayOfWeekInSetNow(String weekSet)`

Availability: RedwoodScript only. Requires 'Module.AdvancedScripting' key.

Is the current day of the week in *weekSet*.

isDayOfWeekInSet

Signature: `boolean Time.isDayOfWeekInSet(Object date, String weekSet)`

Availability: RedwoodScript only. Requires 'Module.AdvancedScripting' key.

Is the day of the week of *date* in *weekSet*.

isDayInRangeNow

Signature: `boolean Time.isDayInRangeNow(String dayRange)`

Availability: RedwoodScript only. Requires 'Module.AdvancedScripting' key.

Is the current day of the month in the range *dayRangeSet*.

isDayInRange

Signature: `boolean Time.isDayInRange(Object date, String dayRange)`

Availability: RedwoodScript only. Requires 'Module.AdvancedScripting' key.

Is the day of the month of *date* in the range *dayRangeSet*.

isLastDayInMonthNow

Signature: `boolean Time.isLastDayInMonthNow(ScriptEvaluationContext context)`

Availability: RedwoodScript only. Requires 'Module.AdvancedScripting' key.

Is it currently the last day of the month?

isLastDayInMonth

Signature: `boolean Time.isLastDayInMonth(Object date)`

Availability: RedwoodScript only. Requires 'Module.AdvancedScripting' key.

Is the date specified the last day of that month?

isMonthInSetNow

Signature: `boolean Time.isMonthInSetNow(String monthRange)`

Availability: RedwoodScript only. Requires 'Module.AdvancedScripting' key.

Is the month of the current date in the range *monthRange*?

isMonthInSet

Signature: `boolean Time.isMonthInSet(Object date, String monthRange)`

Availability: RedwoodScript only. Requires 'Module.AdvancedScripting' key.

Is the month of the specified *date* in the range *monthRange*?

format

Signature: `String Time.format(Object date, String format)`

Availability: RedwoodScript only. Requires 'Module.AdvancedScripting' key.

Format the date according to the specified .

formatDuration

Signature: `String Time.formatDuration(BigDecimal duration)`

Availability: RedwoodScript only. Requires 'Module.AdvancedScripting' key.

Format a duration using the standard format: |1 week, |# weeks, |1 day, |# days, |#|:#|:#|.#

formatDurationEx

Signature: `String Time.formatDurationEx(BigDecimal duration, String format)`

Availability: RedwoodScript only. Requires 'Module.AdvancedScripting' key.

Format a duration using a custom format. The format is specified as: |1 week, |# weeks, |1 day, |# days, |#|:#|:#|.#| The first element is used for 1 week. The second element is used for n weeks, where n > 1. The third element is used for 1 day. The fourth element is used for n days, where n > 1. The fifth element is used for hours. The sixth element is used for minutes. The seventh element is used for seconds. The eighth element is used for milliseconds. In all elements but the first and third, # is replaced with the actual number. If an element is empty it is omitted. If the first element is empty, weeks are not printed and and days may be greater than 7. If the third element is empty, days are not printed and and hours may be greater than 24.

getUTCMillisecondsNow

Signature: `BigDecimal Time.getUTCMillisecondsNow()`

Availability: RedwoodScript only. Requires 'Module.AdvancedScripting' key.

Get the number of milliseconds since midnight on January 1 1970 for the current time.

getUTCMilliseconds

Signature: `BigDecimal Time.getUTCMilliseconds(Object date)`

Availability: RedwoodScript only. Requires 'Module.AdvancedScripting' key.

Get the number of milliseconds since midnight on January 1 1970 for the specified time.

formatNow

Signature: `String Time.formatNow(String format)`

Availability: RedwoodScript only. Requires 'Module.AdvancedScripting' key.

Format the current date according to the specified .

parse

Signature: `DateTimeZone Time.parse(Object string, String format)`

Availability: RedwoodScript only. Requires 'Module.AdvancedScripting' key.

Parse the string into a date according to the specified .

nextTimeWindowOpeningNow

Signature: `DateTimeZone Time.nextTimeWindowOpeningNow(String timeWindow)`

Availability: RedwoodScript only. Requires 'Module.AdvancedScripting' key.

Next opening of *timeWindow* after now.

nextTimeWindowOpening

Signature: `DateTimeZone Time.nextTimeWindowOpening(Object date, String timeWindow)`

Availability: RedwoodScript only. Requires 'Module.AdvancedScripting' key.

Next opening of *timeWindow* after the specified *date*.

nextTimeWindowClosingNow

Signature: `DateTimeZone Time.nextTimeWindowClosingNow(String timeWindow)`

Availability: RedwoodScript only. Requires 'Module.AdvancedScripting' key.

Next opening of *timeWindow* after now.

nextTimeWindowClosing

Signature: `DateTimeZone Time.nextTimeWindowClosing(Object date, String timeWindow)`

Availability: RedwoodScript only. Requires 'Module.AdvancedScripting' key.

Next opening of *timeWindow* after the specified *date*.

Constraint

Prefix: Constraint

Availability: Requires 'Module.Scripting' key.

listConstraint

Signature: `boolean Constraint.listConstraint(String titles, String list, boolean valid)`

Availability: Requires 'Module.Scripting' key.

pairListConstraint

Signature: `boolean Constraint.pairListConstraint(String titles, String list, boolean valid)`

Availability: Requires 'Module.Scripting' key.

JobChainParameters

Prefix: JobChainParameters

Availability: Requires 'Module.Scripting' key.

Job chain job names take the form *Step name*, job *job number*. For example: "Step 2, job 3" or "Data load, job 2".

getOutValueString

Signature: `String JobChainParameters.getOutValueString(String jobName, String parameterName)`

Availability: Requires 'Module.Scripting' key.

Get the value of the String output parameter *parameterName* of the job named *jobName* in the current job chain.

getOutValueNumber

Signature: `BigDecimal JobChainParameters.getOutValueNumber(String jobName, String parameterName)`

Availability: Requires 'Module.Scripting' key.

Get the value of the Number output parameter *parameterName* of the job named *jobName* in the current job chain.

getOutValueDate

Signature: `DateTimeZone JobChainParameters.getOutValueDate(String jobName, String parameterName)`

Availability: Requires 'Module.Scripting' key.

Get the value of the Date output parameter *parameterName* of the job named *jobName* in the current job chain.

getInValueString

Signature: `String JobChainParameters.getInValueString(String jobName, String parameterName)`

Availability: Requires 'Module.Scripting' key.

Get the value of the String input parameter *parameterName* of the job named *jobName* in the current job chain.

getInValueNumber

Signature: `BigDecimal JobChainParameters.getInValueNumber(String jobName, String parameterName)`

Availability: Requires 'Module.Scripting' key.

Get the value of the Number input parameter *parameterName* of the job named *jobName* in the current job chain.

getInValueDate

Signature: `DateTimeZone JobChainParameters.getInValueDate(String jobName, String parameterName)`

Availability: Requires 'Module.Scripting' key.

Get the value of the Date input parameter *parameterName* of the job named *jobName* in the current job chain.

getJobId

Signature: `Long JobChainParameters.getJobId(String jobName)`

Availability: Requires 'Module.Scripting' key.

Get the job id of the job named *jobName* in the current job chain.

getJobStatus

Signature: `String JobChainParameters.getJobStatus(String jobName)`

Availability: Requires 'Module.Scripting' key.

Get the job status of the job named *jobName* in the current job chain.

Query

Prefix: Query

Availability: Requires 'Module.Scripting' key.

Functions for querying the data model.

getNumber

Signature: `BigDecimal Query.getNumber(String query [, Object [] bindVariables])`

Availability: Requires 'Module.Scripting' key.

Get the first column of the first row of the query *query* with optional bind variables bindVariables. Bind variable place holders are specified as ? in the query.

getString

Signature: `String Query.getString(String query [, Object [] bindVariables])`

Availability: Requires 'Module.Scripting' key.

Get the first column of the first row of the query *query* with with optional bind variables bindVariables. Bind variable place holders are specified as ? in the query.

getRelativeJob

Signature: `BigDecimal Query.getRelativeJob(String jobName)`

Availability: Requires 'Module.Scripting' key.

Get the job id of the job named *jobName* in the current job chain.

SAP

Prefix: SAP

Availability: Requires 'Module.Scripting' key.

convertJobParameterRangesToExpression

Signature: `String SAP.convertJobParameterRangesToExpression(String jobName, String parameterName, int fieldLength)`

Availability: Requires 'Module.Scripting' key.

Get the value of the String output parameter *parameterName* of the job named *jobName* in the current job chain. Convert ranges to a selopt expression. Ranges are sent as a long string in the form [<sign><option><low><high>]+ where <sign> is a single character I or E <option> a select option, e.g. EQ, BT <low> the low value of the range <high> the high value of the range All fields are always sent, regardless of the value of the <option>. If the value is shorter than the length of the field, it is padded with blanks. Such ranges are known to be sent by Closing Cockpit.

convertRangesToExpression

Signature: `String SAP.convertRangesToExpression(String rangesString, int fieldLength)`

Availability: Requires 'Module.Scripting' key.

Convert ranges to a selopt expression. Ranges are sent as a long string in the form [<sign><option><low><high>]+ where <sign> is a single character I or E <option> a select option, e.g. EQ, BT <low> the low value of the range <high> the high value of the range All fields are always sent, regardless of the value of the <option>. If the value is shorter than the length of the field, it is padded with blanks. Such ranges are known to be sent by Closing Cockpit.

Table

Prefix: Table

Availability: Requires 'Module.Scripting' key.

Functions for querying a table.

getRow

Signature: `String Table.getRow(String table, String key)`

Availability: Requires 'Module.Scripting' key.

getColumnString

Signature: `String Table.getColumnString(String tableName, String key, String column)`

Availability: Requires 'Module.Scripting' key.

getColumnNumber

Signature: `BigDecimal Table.getColumnNumber(String table, String key, String column)`

Availability: Requires 'Module.Scripting' key.

formatRow

Signature: `String Table.formatRow(String table, String key, String rowStart, String columnFormat, String columnSeparator, String rowEnd)`

Availability: Requires 'Module.Scripting' key.

Variable

Prefix: Variable

Availability: Requires 'Module.Scripting' key.

Functions for querying the System_Variables table.

getString

Signature: `String Variable.getString(String key)`

Availability: Requires 'Module.Scripting' key.

getNumber

Signature: `BigDecimal Variable.getNumber(String key)`

Availability: Requires 'Module.Scripting' key.

JCS Errors

JCS Errors 101XXX (alert)

CODE	Description
JCS-101000	A loop was detected in a graph
JCS-101001	An invalid schema was detected in an address.
JCS-101002	No gateway was found for the specified address.
JCS-101003	Unable to match an address with a pattern.

JCS Errors 102XXX (api-impl)

CODE	Description
JCS-102000	An action has vetoed the transaction.
JCS-102001	An application does not belong to the same partition as its parent.
JCS-102002	When a Job is to be automatically submitted due to a JobDefinitionWaitEvent it must have a default Queue.
JCS-102003	The behavior of the object prevents the action being performed.
JCS-102004	An error occurred trying to create a branched-copy for an object.
JCS-102005	The path for a business key is unavailable.
JCS-102006	Jobchain calls must refer to master job definitions if and only if they belong to a master job definition.
JCS-102007	Jobchain calls cannot refer to template job definitions.
JCS-102008	The isolation-group of a partition was changed.
JCS-102009	It is not possible to delete a process server unless it is stopped.
JCS-102010	Only jobs that are steps or calls can be enabled.
JCS-102011	The Custom Library cannot be renamed.
JCS-102012	The in-value of a parameter was changed on a job has started.
JCS-102013	The out-value of a parameter was changed on a job that was already final.
JCS-102014	Cannot modify the scheduled start time of a Job once it has already started or if it has already completed.
JCS-102015	Submit frames cannot be changed for maintenance jobs.
JCS-102016	Cannot modify an inbuilt role.

JCS-102017	Only system objects can use system job locks.
JCS-102018	Cannot prepare a non-current job definition.
JCS-102019	Cannot prepare a template job definition.
JCS-102020	The requested action cannot be executed on an object that has not been persisted.
JCS-102021	Cannot submit a job prepared from a non-current job definition.
JCS-102022	The job's parent was modified; this is not allowed.
JCS-102023	Job chain steps cannot be restarted.
JCS-102024	System maintenance jobs cannot be restarted.
JCS-102025	Restricted SAP jobs cannot be restarted.
JCS-102026	Job chain steps cannot be resubmitted.
JCS-102027	System maintenance jobs cannot be resubmitted.
JCS-102028	Restricted SAP jobs cannot be resubmitted.
JCS-102029	The Template property cannot be reverted.
JCS-102030	A job definition that is already used as jobchain call cannot be made a template.
JCS-102031	Calendar type time windows define whole days, so times cannot be set.
JCS-102032	Calendar type time windows define whole days, so time zones cannot be set.
JCS-102033	Cannot use the scheduler session when it is in the detached state.
JCS-102034	The database control connection could not be initialized.
JCS-102035	Creator-grants can only be created at the same time as the object to which they apply.
JCS-102036	An object refers to another object that is in a different isolation group.
JCS-102037	Day shift attributes are only allowed for Calendar type time windows.
JCS-102038	The 'StartAtValue' attribute for a job is not allowed.
JCS-102039	Duplicate ABAP client number combination.
JCS-102040	An "early" timeout has occurred before an operation completed successfully.
JCS-102041	No RFC connect string given.
JCS-102042	No J2EE connect string given.
JCS-102043	Too many XBP clients have been defined.
JCS-102044	Too many J2EE clients of a given type have been defined.
JCS-102045	Too many SAP systems have been defined.
JCS-102046	A job-definition has a forbidden related object.
JCS-102047	An attempt has been made to set a field that must be null.
JCS-102048	A security grant has been made by a subject that is not a user.
JCS-102049	A loop has been detected in a series of objects that must form a hierarchy without any loops.
JCS-102050	Inconsistent object types for matches/overrides of an ImportRule.
JCS-102051	A status handler attached to a job chain has an action of IgnoreJobChainDefault.
JCS-102052	The user has tried to do something that they do not have privileges to do.
JCS-102053	The user has been designated as a super reader, and they have tried to call persist. This is not allowed.

JCS-102054	The branch timestamp must be set if (and only if) the object is not the master.
JCS-102055	Users are not allowed to set the RaiserObject or ClearerObject to something else then the current user or a job.
JCS-102056	An event can only be deleted when it is has been cleared.
JCS-102057	The expression cannot be compiled.
JCS-102058	The hold on submit expression is invalid.
JCS-102059	The time window status specified requires a time window.
JCS-102060	The time window status specified requires no time window.
JCS-102061	The specified operation is illegal.
JCS-102062	The specified operation is illegal.
JCS-102063	Transaction vetoed due to license conditions being violated.
JCS-102064	An operation was underway when the thread was interrupted.
JCS-102065	JobDefinitionKeepType is specified and JobDefinitionKeepUnits is not set to job.
JCS-102066	The queue expression is invalid.
JCS-102067	The priority expression is invalid.
JCS-102068	Root exception for reasons a master object reference is invalid.
JCS-102069	An invalid persistence phase was specified.
JCS-102070	Invalid return-code mapping expression.
JCS-102071	The state transition that is trying to be done is not allowed for the current state
JCS-102072	Base class for all exceptions relating to invalid status handlers.
JCS-102073	A global object refers to another object that is isolated.
JCS-102074	An object refers to another object that is not allowed due to isolation.
JCS-102075	Job-files can only be associated with a service if the service supports job-files.
JCS-102076	A job cannot be modified when it has reached a final state status.
JCS-102077	A job finished with an unexpected final status.
JCS-102078	The job is not part of a recurrence group.
JCS-102079	There is no SAP system for this process server.
JCS-102080	Transaction vetoed due to a licensing problem.
JCS-102081	The value provided for the client number was malformed.
JCS-102082	The value provided for the J2EE user name is malformed.
JCS-102083	The process definition to be called on error is invalid.
JCS-102084	A mandatory field is not set.
JCS-102085	A mandatory job parameter was not set.
JCS-102086	Master objects must have themself as master.
JCS-102087	Minimum value must be smaller then maximum value.
JCS-102088	Jobfile component could not be located. All jobfile content is retrieved via this component.
JCS-102089	Specified isolation-group is missing.
JCS-102090	A related object for this type of job definition is missing.
JCS-102091	A reserved job-definition parameter is required but missing.

JCS-102092	Specified user is missing.
JCS-102093	License does not allow model modification.
JCS-102094	Monitored object is does not define a business key.
JCS-102095	Referenced time window must be a Calendar since this time window is a Calendar.
JCS-102096	A job can only be deleted when it is in a final state.
JCS-102097	Non-master objects must refer to the current master.
JCS-102098	No default ABAP client has been defined for the SAP system.
JCS-102099	An attempt was made to update an object, and the corresponding row could not be found in the database.
JCS-102100	An attempt has been made to change a non-modifiable field.
JCS-102101	An attempt has been made to change a non-modifiable object.
JCS-102102	An attempt was made to link a new object to a one to one relationship, but the relationship was already filled.
JCS-102103	Setting the field of an object would introduce an object cycle.
JCS-102104	Olson name of timezone does not match the name.
JCS-102105	The reply to an OperatorMessage was not allowed.
JCS-102106	The ReplyExpression attribute is not a valid regular expression.
JCS-102107	Replies are not allowed for this OperatorMessage.
JCS-102108	The reply to an OperatorMessage is not formatted correctly.
JCS-102109	The reply to an OperatorMessage can only be set once.
JCS-102110	The job chain call has a parameter mapping that refers to a parameter that does not belong to the job definition referred to by the call.
JCS-102111	Baseclass for parameter mapping exceptions
JCS-102112	The target parameter of a JobChainCallOutReferenceParameter must be on the master job chain
JCS-102113	Parameter types of parameter mapping do not match
JCS-102114	Parameter direction of parameter mapping is incorrect
JCS-102115	Order of mapped parameter calls is incorrect
JCS-102116	An attempt is being made to change a partition of an object.
JCS-102117	The parameters are not valid for the period function on a time window element.
JCS-102118	Grant is for a type of object that has no defined privileges.
JCS-102119	Privilege subjects cannot be associated with a specific isolation group.
JCS-102120	Either both or neither of ProcessServer and Service must be specified for job-files.
JCS-102121	The recurrence pattern was changed on a job-group that has no modifiable jobs associated with it.
JCS-102122	An exception related to recurrence actions on a job.
JCS-102123	An attempt has been made to set both a Submit Frame and a Recurrence pattern on this Job.
JCS-102124	The value of the registry entry must match the session user.
JCS-102125	Often an operation cannot be performed if an associated object is not visible.

JCS-102126	Root class for problems with related objects of job definitions.
JCS-102127	Base class for problems relating to a reserved job-definition parameters.
JCS-102128	A parameter definition has a reserved name.
JCS-102129	A parameter definition has a reserved name.
JCS-102130	A reserved job-definition parameter has the wrong type.
JCS-102131	Role-grants cannot be modified; they must be deleted and recreated.
JCS-102132	Grantor does not have the role with grantable rights.
JCS-102133	The related ImportRuleDefinition must support matching.
JCS-102134	The related ImportRuleDefinition must support overriding.
JCS-102135	Base class for exceptions relating to the isolation of SAP systems.
JCS-102136	Isolation of an isolated SAP system would be violated.
JCS-102137	SAP system should be isolated but is not.
JCS-102138	An attempt was made to change the scheduling settings on a job that has a recurrence pattern.
JCS-102139	The scheduling parameter evaluated to a queue that does not exist.
JCS-102140	The routing parameter evaluated to a process server that does not exist.
JCS-102141	The routing parameter evaluated to a process server that does not have any queues.
JCS-102142	An error was encountered while evaluating and/or applying a scheduling parameter.
JCS-102143	An error was encountered while evaluating a scheduling parameter.
JCS-102144	Internal error obtaining script engine for expression.
JCS-102145	An attempt was made to call setDirect on a sealed object.
JCS-102146	Custom non-rank privileges require advanced-security.
JCS-102147	Base-class for exceptions relating to security grants.
JCS-102148	Grant created at a level that is not appropriate for the given level.
JCS-102149	Base-class for exceptions relating to security privilege grants.
JCS-102150	Grantor does not have rights to modify the specified privilege.
JCS-102151	The next value in the sequence could not be retrieved for some reason.
JCS-102152	Sequence could not be created and/or initialized for some reason.
JCS-102153	Root class for runtime exceptions related to sequences.
JCS-102154	The ProcessServerService that is being deleted is needed by the ProcessServer to handle a JobDefinitionType that is still defined for the ProcessServer.
JCS-102155	Process server services can only be added or deleted when the process server is not running.
JCS-102156	Operation could not be performed because an object is no longer current.
JCS-102157	A status handler has an attribute that is forbidden for the given handler action.
JCS-102158	A status handler is missing a required attribute for the given handler action.
JCS-102159	The string provided for the given field is too long.
JCS-102160	A security grant has been given to a subject that cannot receive grants.
JCS-102161	A timeout expired before an operation completed successfully.
JCS-102162	Too many default ABAP client have been defined for the SAP system.

JCS-102163	Job must be assigned to a process-server.
JCS-102164	More than one object shares the same sequence number.
JCS-102165	Job cannot be restarted for some unknown reason.
JCS-102166	Job cannot be resubmitted for some unknown reason.
JCS-102167	Transaction vetoed due to a feature not being licensed.
JCS-102168	Transaction vetoed due to neither of two features being licensed.
JCS-102169	The value provided for the given field is not contained in the list of values for this field./
JCS-102170	The value provided for the given field is too large./
JCS-102171	The value provided for the given field is too small
JCS-102172	A forbidden value field for this parameter has been set.
JCS-102173	A forbidden value field for this parameter type has been set.
JCS-102174	A forbidden value field for this parameter direction has been set.
JCS-102175	Only chained jobs can be disabled.
JCS-102176	Holding a job in is forbidden based on its current status.
JCS-102177	Killing of jobs that are jobchain steps is not possible.
JCS-102178	Only executing jobs can be killed.
JCS-102179	Olson name is not known to system.
JCS-102180	Name of timezone matches another timezone's Olson name.
JCS-102181	Mandatory job parameter for call missing mapping.
JCS-102182	Job actions are not licensed.
JCS-102183	Invalid action source.
JCS-102184	The behavior of the object prevents the action being performed.
JCS-102185	The table must have a mandatory column called Key
JCS-102186	The specified column for this table is not nullable
JCS-102187	The specified column for this table is a number
JCS-102188	The specified column for this table is a date
JCS-102189	The specified column is not allowed for this table
JCS-102190	Data is required for a simple constraint.
JCS-102191	Data is required for a simple constraint.
JCS-102192	Renaming of the object is not permitted when attached to a SAP system.
JCS-102193	A BeforeDefinitionChange Trigger has vetoed the transaction.
JCS-102194	Compilation failed.
JCS-102195	An unexpected compiler exception occured.
JCS-102196	Match compilation failed.
JCS-102197	Persist not allowed.
JCS-102198	Two fields are set that conflict with each other.
JCS-102199	The specified audit level is not allowed for this object type
JCS-102200	ReportColumn Object Definition does not match the ObjectDefinition of the Report.

JCS-102201	Exception for when the overdue action (change status, raise event) is invalid. Either an status or an event to raise must be defined.
JCS-102202	The check on the RELEntryPoint has failed .
JCS-102203	Simple constraints cannot be applied to this data type.
JCS-102204	The Object Definition for the Audited object was not defined.
JCS-102205	The System Process Server cannot be a Process Server resource.
JCS-102206	Invalid method signature for RELEntryPoint.
JCS-102207	Attribute on a time window element is out of bounds and does not match the special value for "last".
JCS-102208	Isolation group creation prevented by license.
JCS-102209	Maximum number of retries for branching the job definition was exceeded.
JCS-102210	File component could not be located. All file content is retrieved via this component.
JCS-102211	The transaction has been vetoed because it contains a non-unique order value.
JCS-102212	An isolated SAP system can't be created in the GLOBAL partition.
JCS-102213	An isolated SAP system can't be created because two or more clients defined for this SAP system.
JCS-102214	An isolated SAP system can't be created because non-standard services exists in the current configuration.
JCS-102215	Job chain calls may not have a submit frame.
JCS-102216	Job chain calls may not have a recurrence.
JCS-102217	Job chain calls may not have a recurrence.
JCS-102218	An isolated SAP system can't be created another SAP system is already found in this isolation group.
JCS-102219	The value provided for the client interface version was malformed.
JCS-102220	The field that the user tried to change is restricted.
JCS-102221	The detail that the user tried to change is restricted.
JCS-102222	Query Filter Simple constraints can only be applied to the String data type.
JCS-102223	An attempt was made to save an object into a different isolation group than the one that the user is currently logged in with.
JCS-102224	A step or call can only be deleted when its parent is in a final state.
JCS-102225	A step or call can only be deleted when its parent can be deleted.
JCS-102226	An attempt was made to delete the inbuilt custom library.
JCS-102227	A wrapped job can only de deleted if the corresponding wrapped job can be seen.
JCS-102228	A wrapped job can only de deleted if the corresponding wrapped job can be deleted.
JCS-102200	An attempt was made to store an object in the database that would have violated uniqueness constraints.
JCS-102201	An attempt was made to delete an object that is still in use by another object.
JCS-102260	Wrapper for ForeignKeyViolationVetoWrapperExceptions.

JCS Errors 103XXX (api-int)

CODE	Description
JCS-103000	An action was performed on an object that is not possible.
JCS-103001	This indicates that an error has occurred while trying to perform a query.
JCS-103002	Cancellation of jobs that are jobchain steps is not allowed.
JCS-103003	Jobs can only be canceled prior to execution.
JCS-103004	A request was made for a file from a job, and the file with the required name did not exist.
JCS-103005	Root exception for reasons a job parameter is invalid.
JCS-103006	A request was made for a file from a job, and the file with the required name did not exist.
JCS-103007	A request was made for a job, however the job did not exist.
JCS-103008	An exception occurred with a parameter.
JCS-103009	An error was encountered while retrieving a parameter mapping value.
JCS-103010	A parameter was not found.
JCS-103011	No value was specified for a required parameter.
JCS-103012	No value was specified for a required parameter.
JCS-103013	An invalid value was specified for a parameter.
JCS-103014	The parameter contains invalid substitution variables.
JCS-103015	Restarting this job is prohibited.
JCS-103016	Resubmitting this job is prohibited.
JCS-103017	Checker implementation expects key of a specific type.
JCS-103018	You cannot delete your own isolation group.
JCS-103019	The current user has no isolation group.

JCS Errors 104XXX (api)

CODE	Description
JCS-104000	A communal scheduler session factory has not been published yet.
JCS-104001	An exception was thrown while processing the callback.
JCS-104002	Base class for exceptions relating to exporting objects.
JCS-104003	Unable to search job file because of a permanent remote exception.
JCS-104004	Search failed for a some reason.
JCS-104005	Unable to search job file because of a remote exception.
JCS-104006	Remote login failed.
JCS-104007	Unable to list directory because of a permanent remote exception.
JCS-104008	List directory for some reason.
JCS-104009	Unable to list directory because of a remote exception.
JCS-104010	The specified reply was invalid.

JCS Errors 105XXX (as400)

CODE	Description
JCS-105000	AS400 connection exception.
JCS-105001	AS/400 security violation exception.
JCS-105002	AS/400 job failed exception.
JCS-105003	AS/400 invalid parameter exception.
JCS-105004	AS/400 event exception.

JCS Errors 106XXX (audit)

CODE	Description
JCS-106000	The transaction requires a reason to be defined to explain the modification.

JCS Errors 107XXX (cl-api)

CODE	Description
JCS-107000	Duplicate component registration exception.
JCS-107001	Registration currently forbidden.
JCS-107002	The component manager is currently unavailable. This is normally a temporary problem.
JCS-107003	A communal scheduler session factory has not been published yet.

JCS Errors 108XXX (cl-impl)

CODE	Description
JCS-108000	Registration of a component was attempted after the component lifecycle commenced.
JCS-108001	JNDI bindings failed.
JCS-108002	No backend could retrieve the jobfile; all backends reported a permanent failure.
JCS-108003	No backend could retrieve the jobfile, but at least one backend reported a transient failure. This indicates that the same request might succeed in the future.
JCS-108004	An IO exception occurred accessing a local file.
JCS-108005	An IO exception occurred accessing a jobfile.
JCS-108006	An IO exception occurred accessing a local file.
JCS-108007	An IO exception occurred accessing a file.
JCS-108008	No backend could retrieve the file; all backends reported a permanent failure.
JCS-108009	No backend could retrieve the file, but at least one backend reported a transient failure. This indicates that the same request might succeed in the future.
JCS-108010	The Scheduler has not been configured and has no JNDI configuration.

JCS Errors 109XXX (cluster)

CODE	Description
JCS-109000	This exception is thrown when a JNDI lookup succeeds, but the object returned is not of the expected type.
JCS-109001	Lookup of an object returned null.

JCS Errors 110XXX (configuration)

CODE	Description
JCS-110000	Base class for all configuration-related checked exceptions.
JCS-110001	Base class for all configuration-related unchecked exceptions.
JCS-110002	This is the root exception for all problems relating to a configuration entry.
JCS-110003	This is the root exception for all problems relating to a configuration entry that should have been detected earlier.
JCS-110004	This is the root exception for all problems relating to the value of a configuration entry.
JCS-110005	Missing configuration entry.
JCS-110006	Configuration entry should be a number, but does not appear to be.
JCS-110007	Value of configuration entry is not within the acceptable range.
JCS-110008	Value of configuration entry is too high.
JCS-110009	Value of configuration entry is too low.
JCS-110010	Class given for configuration value cannot be loaded.
JCS-110011	Class given for configuration value does not inherit from the correct type.
JCS-110012	Value of configuration entry is not in allowed list.
JCS-110013	Value of boolean entry is invalid.
JCS-110014	Configuration is not an enumeration, but was accessed as one.
JCS-110015	Value of configuration entry is too long.

JCS Errors 111XXX (core)

CODE	Description
JCS-111000	Problem with service class.
JCS-111001	Service class could not be located.
JCS-111002	Service class could not be instantiated.
JCS-111003	A running instance of expected service could not be located.
JCS-111004	A job is submitted on a Queue that has no ProcessServer that provides Resource, JobDefinitionType and Service that are required to run the Job.
JCS-111005	One of the handler services has received a message type that it does not know how to handle.
JCS-111006	The Job Router does not know about an object that the Job is trying to route onto.
JCS-111007	The process server has no services.

JCS-111008	The Job Router could not load a required related object from the database.

JCS Errors 112XXX (db-init)

CODE	Description
JCS-112000	The system can not run the system job because the user does not have the appropriate privileges.
JCS-112001	The system_isolation_setuser job could not find the user in the database.
JCS-112002	The system can not run the system_isolation_setuser job because the registry entry could not be found.
JCS-112003	An application could not be located.
JCS-112004	An export rule-set could not be located.
JCS-112005	An expected input job-file could not be found.
JCS-112006	An object that needs to be upgraded could not be found.
JCS-112007	A system job definition could not be located.
JCS-112008	An error occurred ensuring that the object and field definitions are initialized correctly.
JCS-112009	A field definition already exists but is a reference to the wrong type of object.
JCS-112010	A field definition already exists but is of the wrong type.
JCS-112011	An definition is required, but was marked for deletion.
JCS-112012	This is a special exception, that will only be thrown by the db-init component, and only during the configure phase.
JCS-112013	This exception indicates that one of the epoch initializers has failed.
JCS-112014	Existing object definitions were found that should not be present.
JCS-112015	Existing object field definitions were found that should not be present.
JCS-112016	An object related to a job-definition has been encountered but is not handled.
JCS-112017	An import cannot complete due to unresolved references.
JCS-112018	The wrong number of objects was found.
JCS-112019	An invalid mode was specified for the upgrade job.
JCS-112020	Not all job-files were cleanly deleted.
JCS-112021	An import rule-set was specified that could not be located.
JCS-112022	An object that needs to be upgraded could not be found.
JCS-112023	Base exception class for all Schema Initialization exceptions.
JCS-112024	The database schema could not be located in the database and the system is not configured to perform self initialization.
JCS-112025	An exception occurred when trying to upgrade the database schema.
JCS-112026	The Open SQL schema is out of date. The Schema Initialization component will never automatically update OpenSQL databases.
JCS-112027	The database schema is out of date and the scheduler is not configured to self-upgrade.

JCS Errors 113XXX (export)

CODE	Description
JCS-113000	A model reference cannot be read because it defines no business key.
JCS-113001	Relation {0} on {1} refers to {5}, which defines no business key.
JCS-113002	Reading was attempted for a type of object where this is not supported.
JCS-113003	Writing was attempted for a type of object where this is not supported.
JCS-113004	An unknown field was set for this type of object.
JCS-113005	The parent field could not be set using the supplied business key.
JCS-113006	The field could not be set using the supplied value.
JCS-113007	The field could not be set using the supplied value.
JCS-113008	A nested object of the given type is not known.
JCS-113009	Business key not for expected type of object.
JCS-113010	An object with the same business key is already registered.
JCS-113011	Resolving a business key failed for some reason.
JCS-113012	The object corresponding to this business key does not exist.
JCS-113013	The object for this business-key cannot be resolved without first resolving other objects, but they in turn depend require this business-key.
JCS-113014	An unknown error occurred during archival.
JCS-113015	An attempt was made to export an object where this is not supported.
JCS-113016	An attempt was made to archive an object that does not define a business key.
JCS-113017	Base class for exceptions relating to importing objects.
JCS-113018	An exception occurred while trying to parse the import source.
JCS-113019	An import was performed, but no object was defined in the process.
JCS-113020	A written object does not have the business key it was declared to have.
JCS-113021	A object being created does not match the type of its declared business key.
JCS-113022	An object being created should replace an existing object, but the existing object could not be loaded.
JCS-113023	An object is being replaced by the model writer, but has a different parent to the existing object.
JCS-113024	An override was applied to an object of the wrong type.
JCS-113025	An override was applied to a job definition that has no script.
JCS-113026	An override was applied with a null value for a RuleDefinition that needs this value.
JCS-113027	An parameter override was applied where a parameter with the supplied name could not be found on the object.
JCS-113028	An import action failed to execute when run as part of the override clauses.
JCS-113029	An object was used, but it needs to be defined in the same session.
JCS-113030	Two parent objects maintain the same collection as nested objects but are not consistent.
JCS-113050	Multiple keys could not be found during an import.

JCS Errors 114XXX (infrastructure)

CODE	Description
JCS-114000	Root exception for all exceptions where the request cannot be understood.
JCS-114001	A portion of the URL is malformed and cannot be parsed.
JCS-114002	The value of a component in the URL cannot be parsed.
JCS-114003	Request is missing information.
JCS-114004	Request is empty; did not contain any information.
JCS-114005	A component was encountered that was not expected; either it's not allowed, or is a duplicate.
JCS-114006	A character encoding was missing from the JVM.
JCS-114007	A problem occurred relating to locating and reading the initial translation data.
JCS-114008	Translation data for the given language could not be location.
JCS-114009	Translation data for the given language could not be read.
JCS-114010	An exception occurred during XML serialization.
JCS-114011	Contents of SLIM file are invalid.
JCS-114012	A specified key in a SLIM file is not allowed.
JCS-114013	A zero-length (empty) SLIM key was encountered.
JCS-114014	Translation text is invalid.
JCS-114015	Incomplete escape-sequence encountered at the end of translation text.
JCS-114016	Invalid escape-sequence encountered in translation text.
JCS-114017	The JNDIComponent has not be initialized and JNDILookup's cannot be performed.
JCS-114018	General secure-storage exception.

JCS Errors 115XXX (jobchainservice)

CODE	Description
JCS-115000	A new iteration of a step could not be created.

JCS Errors 116XXX (jobservice)

CODE	Description
JCS-116000	A platform agent could not be initialized because its already in use by another process-server..
JCS-116001	The Job Runner could not be instantiated for some reason.
JCS-116002	The Job Runner has had an error when executing the job.
JCS-116003	An error occurred while processing the process server parameters while starting the service.
JCS-116004	The Job Runner could not create an output file
JCS-116005	An error occurred during auto update
JCS-116006	The latest version of the installer for the platform could not be found.

JCS-116007	The platform agent is not configured for, or does not support, auto update.
JCS-116008	The platform agent update failed.
JCS-116009	The platform agent restart failed.
JCS-116010	The platform agent returned an HTTP response code indicating failure.
JCS-116011	Failed to connect to platform agent; the network returned an error.
JCS-116012	Failed to connect to platform agent; the network returned an error. The message is classified as transient (we hope it will go away the next time).

JCS Errors 117XXX (mailservice-api)

CODE	Description
JCS-117000	mail invalid parameter exception.
JCS-117001	No mail server defined.

JCS Errors 118XXX (security)

CODE	Description
JCS-118000	Cannot instantiate any HTTP authentication provider.
JCS-118001	Security exception
JCS-118002	Exception creating a user
JCS-118003	Exception creating a user
JCS-118004	Exception importing security
JCS-118005	Exception importing a user
JCS-118006	Exception creating a role
JCS-118007	Exception creating a role
JCS-118008	Exception importing a role
JCS-118009	No such isolation group exception
JCS-118010	Invalid SystemId

JCS Errors 119XXX (servlet)

CODE	Description
JCS-119000	Root exception for all reasons a jobfile cannot be located.
JCS-119001	There is no known job-file that has the given unique identifier.
JCS-119002	There is no known log-file for a Trigger that has the given unique identifier.
JCS-119003	There is no known Trigger that has the given unique identifier.
JCS-119004	There is no known job that has the given identifier.
JCS-119005	There is no job-file from a job with the given file-number.
JCS-119006	The specified object cannot be located.
JCS-119007	A session could not be created because a user could not be associated with it.

JCS Errors 120XXX (ui)

CODE	Description
JCS-120000	Base class for UIExceptions.
JCS-120001	Base class for concurrent update exceptions.
JCS-120002	The requested object is no longer available, it has either been deleted or the security settings prevent viewing the object.
JCS-120003	Illegal type of the requested object.
JCS-120004	The linked object is no longer available, it has either been deleted or the security settings prevent viewing the object.
JCS-120005	A mandatory field is not being set.
JCS-120006	The object that needs to be updated has been modified by another user.
JCS-120007	The object that needs to be acted upon does not have the named action.
JCS-120008	The data could not be decoded to the desired type.

JCS Errors 121XXX (module-r2w)

CODE	Description
JCS-121000	The configuration could not be found to contact the watcher.

JCS Errors 122XXX (agent-api)

CODE	Description
JCS-122000	Root exception for all exceptions where the request cannot be understood.
JCS-122001	The call is not supported.

JCS Errors 122XXX (model)

CODE	Description
JCS-122000	While trying to submit a job one of the constraints has been violated.
JCS-122001	A day has been entered that is out of the specified allowed range.
JCS-122002	The dirty list on the session is not empty, therefore the requested operation cannot be completed
JCS-122003	The database is an inconsistent state since a duplicate primary key was found.
JCS-122004	Exceptions derived from here indicate a failure to retrieve the file content that is likely permanent in nature.
JCS-122005	Exceptions derived from here indicate a failure to retrieve the file content that is likely transient in nature; a retry may succeed.
JCS-122006	A user that has been marked as inactive cannot login.
JCS-122007	An attempt was made to create more objects then the license allows.
JCS-122008	The value for the given key is not valid.
JCS-122009	The name of the object is invalid

JCS-122010	An operation has been attempted that requires that the Job has a parent, but it does not.
JCS-122011	Exceptions derived from here indicate a failure to retrieve the jobfile that is likely permanent in nature.
JCS-122012	Exceptions derived from here indicate a failure to retrieve the jobfile that is likely transient in nature; a retry may succeed.
JCS-122013	Exceptions derived from here indicate a license violation
JCS-122014	Exceptions derived from here indicate a failure to retrieve the directory content that is likely permanent in nature.
JCS-122015	Exceptions derived from here indicate a failure to retrieve the directory content that is likely transient in nature; a retry may succeed.
JCS-122016	Only clean JobDefinitions can be prepared. This exception will be thrown when prepare has been called on a JobDefinition that has been modified since being retrieved from the database, or that has not yet been persisted to the database.
JCS-122017	No timezone could be found to calculate this TimeWindowElement. This means that the object that includes the TimeWindow should set the TimeZone to use for the calculation of the TimeWindow, or a TimeZone must be set on the TimeWindow or TimeWindowElement.
JCS-122018	The object could not be created since it would violate the object constraints.
JCS-122019	The object has been marked for deletion, no operations are allowed on the object anymore.
JCS-122020	The object is already on the dirtylist in another subsession.
JCS-122021	The object is not attached to a session, and cannot be used anymore.
JCS-122022	The object used for assignment is not from the same session as the object it is being assigned to
JCS-122023	An import override rule could not be applied to an object.
JCS-122024	A Parameter was set, but using the wrong type of data.
JCS-122025	An error was encountered while evaluating a default parameter value.
JCS-122026	An attempt has been made to raise an event that has isPendingEventsAllowed==false, and the event is already raised.
JCS-122027	Internal error in a period function.
JCS-122028	Period functions should implement the PeriodFunctionCalculator interface.
JCS-122029	Unable to configure and/or retrieve a low-level persistence factory.
JCS-122030	Error during computing recurrence.
JCS-122031	Error in the recurrence pattern.
JCS-122032	An error was encountered while evaluating the return-code mapping.
JCS-122033	Baseclass for checked exceptions thrown by the API.
JCS-122034	Unable to persist.
JCS-122035	Invalid string literal.
JCS-122036	The string passed in does not represent a valid ObjectType.
JCS-122037	Invalid enumeration value.
JCS-122038	The persist call did not update any rows in the database.
JCS-122039	Unable to persist.
JCS-122040	An exception occurred during the execution of a persistence layer method.

JCS-122041	Baseclass for unchecked exceptions thrown by the API.
JCS-122042	The scheduler API is currently unavailable. This is normally a temporary problem.
JCS-122043	An error has occurred while trying to configure a Scheduler component.
JCS-122044	An error was encountered while evaluating and/or applying a scheduling parameter.
JCS-122045	An error was encountered while evaluating and/or applying a scheduling parameter.
JCS-122046	An error occurred calculating the opening or closing of a time window.
JCS-122047	Attribute on a time window element is out of bounds.
JCS-122048	CalculateNextClose should only be called on an open time window.
JCS-122049	Time window {0} cannot be called with a NULL opening time.
JCS-122050	An attempt was made to use an unlicensed feature.
JCS-122051	The object that was passed in is of an incorrect type.
JCS-122052	The report generation failed.
JCS-122053	ProcessServers can never have multiple browsable ProcessServerServices.

JCS Errors 123XXX (persistence)

CODE	Description
JCS-123100	An attempt was made to update an object, however it failed, either because the object has been deleted, or because it has been previously modified.
JCS-123101	An error was encountered while trying to parse the SQL statement.

JCS Errors 124XXX (script-java)

CODE	Description
JCS-124000	Validation of the RELEntryPoint failed because the method signature or class could not be found.
JCS-124001	Classes must extend the predefined Stub class where the Stub class it he name of the class with Stub appended to it.
JCS-124002	Classes must extend the predefined Stub class where the Stub class it he name of the class with Stub appended to it.
JCS-124003	The source cannot define an abstract class.
JCS-124004	No source has been defined.
JCS-124005	The import definition in the source is not valid.
JCS-124006	Source has not be correctly started. It must start with a {, import or package statement.
JCS-124007	Importing or accessing classes is restricted to a predefined safe list of classes.
JCS-124008	Importing or accessing classes is restricted to a predefined safe list of classes.
JCS-124009	Line number mapping failed with a duplicate end token.
JCS-124010	Line number mapping failed with unended token.
JCS-124011	Line number mapping is not allowed.
JCS-124012	Line number mapping is not open.

JCS-124013	Line number mapping failed with a end token that does not match the range description.
JCS-124014	No class has been defined in the source
JCS-124015	Source has not be correctly terminated. It must end with a }.

JCS Errors 125XXX (constraint)

CODE	Description
JCS-125000	An attempt was made to submit a job where the parameter was longer than allowed.

JCS Errors 126XXX (iphone)

CODE	Description
JCS-126000	The reply for an operator message was not in the expected format.
JCS-126001	An IO error occurred while receiving the reply for an operator message.
JCS-126002	The intended reply for an operator message could not be determined.
JCS-126003	A push subscription could not be found.

JCS Errors 130XXX (apns)

CODE	Description
JCS-130000	The APNS proxy returned an error.
JCS-130001	The APNS proxy reported that push notifications are disabled for a subscription ID.
JCS-130002	Could not locate key-store with certificate to contact APNS proxy.

JCS Errors 131XXX (dblayer-sql)

CODE	Description
JCS-131000	Failed to upgrade the database. The upgrade operation is not supported.

Glossary

Use

A

ABAP [SAP]

ABAP is a programming language that the SAP R/3 software and later most other software by SAP AG was written in.

ABAP program [SAP]

A program written in the ABAP language.

Application

An object used to group objects like job definitions and job chains.

AS/400 [IBM]

In SAP CPS the older names AS/400 and OS/400 are used to refer to the hardware and operating system as these are the names that have been in use longest. There are at least two other names for the machine:

- From 2000-2004 the machine was known as the IBM eServer iSeries (generally abbreviated as iSeries).
- Since 2004 the machine has been known as the IBM i5 and the operating system as i5/OS.

B

Batch Input Sessions [SAP]

A special SAP job that processes incoming data. A directory is designated in SAP as the Batch Input Folder. Applications can send data, in a special SAP format called iDocs, to the SAP System by placing the files in this directory. The ABAP program RSBDCSUB reads the batch input folder and submits a new SAP job for every file it finds.

C

CCMS [SAP]

The SAP Computer Center Management System (CCMS) is part of an SAP solution that provides common functionality such as user management, monitoring and background jobs.

CCMS Job [SAP]

A background request in CCMS for the execution of one or more ABAP programs and/or repeating system commands.

CEN [SAP]

The SAP monitoring architecture assumes that a system in your system landscape is the central monitoring system (CEN). The monitoring information for your system landscape is collected by this system and displayed in the Structure Alert Monitor.

Client [SAP]

In SAP terminology, a self contained unit in an R/3 system with separate master records and its own set of tables. An SAP Mandanten identified by a 3 digit (numeric) field in SAP CPS.

Constraints

Constraints are conditions that parameters must fulfill.

Context Menu

A list of context sensitive actions that appears by right-clicking. Also called pop-up menus, speed menus, quick menus or shortcut menus.

E

EAR

Enterprise Application Archive. A file that contains a Web-based application. An EAR file is an archive of MIME type "application/zip" and can be extracted with any pack program. It always contains a file in which all of the components belonging to the archive are described. Further applications are necessary to enable the correct implementation of the applications contained in the archive after extraction.

Endpoint

An endpoint is a device that is connected as part of a computer network.

Event

Jobs can wait for one or more events to happen before they start. SAP CPS has two types of events: job events (raised by jobs having specific job statuses or return codes) and file events (raised by files that are created or modified on the operating system). File events require the Platform Agent Module, please check your license.

F

Format

A format can be assigned to a job definition. The format defines the type of output that the job generates or captures.

G H I

GRMG [SAP]

You can use the Generic Request and Message Generator (GRMG) to monitor the availability of technical components and entire business processes. GRMG is suitable both for technical monitoring and for application monitoring.

i5/OS [IBM]

see AS/400

Instance [SAP]

An administrative unit that groups components of an SAP System. The SAP System identification.

iSeries [IBM]

see AS/400

Isolation Group

Isolation groups are used to isolate objects from specific users. Some users can access all isolation groups while others can only access one or more; this is controlled with roles. In the Free of Charge version, isolation groups isolate SAP client/instance combinations from each other allowing you to interact with only one at any given time.

J

Job

A job is a job definition that is submitted to the process server. When users request the execution of a script, a process server creates a job record in the jobs table, calculates the start time and set the job status to SCHEDULED.

Job Chain

A job chain is a special job definition that has no command text and executes one or more job definitions (job chain jobs) in parallel or sequentially. The job chain jobs are grouped into steps. All job chain jobs in one step are executed directly (parallel processing) and process server waits until all job chain jobs get a final job status before proceeding with a next step (sequential processing).

Job Count [SAP]

A job count is a number that represents the id of a job in an SAP System. It is sometimes referred to as *Job Id* in some SAP Systems.

Job Definition

A job definition defines a process to be executed by the process server. A job definition consists of the command text to be executed and several runtime and scheduling controls.

Job Execution

SAP CPS determines when a job can be run according to the specified start date, time window, and so on. A job is executed in a queue. When the start time arrives, the process server creates a runner and executes the job.

Job Id [SAP]

A job id is a number that represents the id of a job in an SAP System. It is sometimes referred to as **Job Count** in some SAP Systems. These is also a job id in SAP CPS, which is SAP CPS specific and is the *unique* identifier of a job in an instance of SAP CPS.

Job Status

Indicates the current condition of a job, such as `Scheduled`, `Never`, `Queued`, `Hold`, `QueueHold`, `Overdue`, `EventWait`, `LockWait`, `Running`, `Completed`, `Console`, `Waiting`, `Deleting`, `Error`, `Canceled`, or `Killed`.

JXBP

Java External Background Processing is an interface to the JAVA stack of an SAP NetWeaver AS. It offers scheduling capabilities for Message-Driven Beans.

K

Keep Clause

The conditions used to determine how many versions of the job output should be kept.

Keep Until

Keep Until is used to specify how long trace files for a specific process server should be kept. The process server checks every 15 minutes for trace files that should be deleted.

L

Lock

Locks are used to make sure that specific job definitions do not run concurrently. You can add multiple locks to a job definition to define which jobs can and cannot run together.

M

Mass Activities [SAP]

A special set of SAP transactions are used to define mass activities. A mass activity run results in a moderate number of jobs can be processed in parallel. This allows good resource usage of large computer systems with multiple CPUs, but avoids the large overheads of too many jobs.

Module

SAP CPS provides a number of ERP interfaces, utilities and systems tools. These are called Modules.

MVS [IBM]

See z/OS.

N O

Node

A node is a device that is connected as part of a computer network. A monitor node is an element of the monitor tree. A leaf node in the monitor tree is a node that does not have any children.

OS/390 [IBM]

See z/OS.

OS/400 [IBM]

See AS/400.

P Q

PI [SAP]

The PI interface allows you to connect to NetWeaver AS Java systems and control communication channels.

Precondition Functions

A precondition is a Boolean Redwood Expression Language function stored in the repository. If the outcome of the function evaluates to `true`, SAP CPS proceeds with the step or job chain job; if the outcome evaluates to `null` or `false`, SAP CPS skips the step or job chain job and sets the job status to `Skipped`.

Process Chain

A sequence of processes that are scheduled in the background to wait for a particular event. Some of these processes trigger an additional event, which in turn can start other processes.

Process Server

Process servers are used to execute jobs.

Queue

Jobs are executed in queues. A queue can be open or held. While a queue is held, no new jobs are started in the queue.

R

Recurrence

Recurrence is used to request multiple executions of a script. There are two types of job recurrence, structured recurrence and calendar recurrence. Structured recurrence is based on a pattern, and calendar recurrence is based on calendar dates.

Rescheduling

Restarting any job that has a final status (such as `Completed`, `Error`, or `Canceled`) or changing the properties of a scheduled or queued job (such as its start time).

Resource

A process server provides one or more resources, such as an Oracle database, an HP LaserJet printer, or a Windows environment. By assigning these resources to the job definition definitions, all jobs for these job definitions are executed by the appropriate process servers.

S

SCH [SAP]

The Business Information Warehouse - Job Scheduling (BW-SCH) interface allows other systems to schedule Process Chains and InfoPackages in SAP BW.

Scheduler

The scheduler is used to refer to the SAP CPS server, usually as *Central Scheduler*.

Scheduling Parameters

A job definition contains a queue, status on submit, and priority. These scheduling parameters define how the process server handles jobs after they are submitted.

Submit Frame

A submit frame is a cyclic calendar which forces a job to be executed respectively. When job definition execution is requested and a submit frame is used, the corresponding job will be restarted automatically by a process server according to the frequency defined in the submit frame.

T

Time Window

A time window defines a time zone in which the job is allowed to be executed. A job calendar that defines the period of time during which a job or a queue can be active. A time window consists of time intervals with either fixed run dates and times or rules such as the first working day of the month, or a combination of run dates and times, and rules. A time window is open whenever one of the time elements of this time window is open.

Time Zones

Time zones are areas of the Earth that have adopted the same standard time. All time zones are defined relative to Coordinated Universal Time (UTC). Time zones are generally centered on meridians of a longitude (a multiple of 15 degrees) which makes neighboring time zones one hour apart. The one hour separation is not always universal, and time zone shapes can be quite irregular because they usually follow the boundaries of states or countries.

U V W X Y Z

Variant [SAP]

A variant is a named object containing a set of valid parameter values that an ABAP program requires to do its work. Only ABAP programs that require parameters require variants. The idea behind a variant is that the parameter values change less often than batch jobs are run. In other words, they can be re-used. The variant concept also allows for extreme task differentiation, where one person determines the contents of the variant, and another person does the scheduling of the ABAP program with the variant.

XAL [SAP]

The External Interface for Alert Management (BC-XAL or XAL) is an interface to the SAP CCMS monitoring infrastructure. It allows other systems to retrieve monitor information from the SAP System.

XBP [SAP]

External Background Processing (BC-XBP) is an interface to the SAP *CCMS Job Scheduling and Event Handling System*. It allows other systems to submit, monitor, and intercept jobs as well as raise and clear events in an SAP System.

XMW [SAP]

External Monitoring Write (BC-XMW or XMW) is an interface to the SAP CCMS monitoring infrastructure. It allows other systems to create, edit, and update SAP Monitors.

z/OS [IBM]

The software platform running on IBM Mainframe computers, currently named *IBM System z*. z/OS has had many names over the years, including MVS and OS/390.

zSeries [IBM]

The name of the hardware platform, generally referred to as Mainframe, has been named eServer zSeries, then zSeries for short.

Attributions

This product may include software developed by third parties, that require attribution or mention of the license under which the original software was distributed. Some third party software is only included in particular editions of the product.

You can find the edition of the product by going to 'Product Information' on the main page (indicated by a small i on the far right hand side on the second row of the screen). Click on this button and look at the Version: field. If this contains (Server) you have the 'Server' edition.

!! OpenSSL

This library is not shipped with the software but can be obtained from Redwood Software on request.

Apache Log4j 1.2.9

This is included only in the 'Server' edition.

This product includes software developed by the Apache Software Foundation (http://www.apache.org/).

The Apache Software License version 1.1 is included as: j2ee/cluster/global/tomcat/lib/rw-apache-logging-log4j-1.2.9-LICENSE.txt

Apache Geronimo 1.0

This is included only in the 'Server' edition.

Redwood Platform 6.0.10

This is included only in the 'Server' edition.

The NOTICE file is included as: j2ee/cluster/global/tomcat/NOTICE

Apache Derby 10.2.1.6

This is included only in the 'Server' edition.

The NOTICE file is included as: j2ee/cluster/global/tomcat/lib/rw-apache-derby-10.2.1.6-NOTICE

Sun JavaMail 1.2

This is included only in the 'Server' edition.

Sun Java Runtime Environment 1.6.0.03 (Win32)

This is included only in the 'Server' edition.

```
DO NOT TRANSLATE OR LOCALIZE.

%% The following software may be included in this product: CS CodeViewer
v1.0; Use of any of this software is governed by the terms of the
license below:

Copyright 1999 by CoolServlets.com.

Any errors or suggested improvements to this class can be reported as
instructed on CoolServlets.com. We hope you enjoy this program... your
comments will encourage further development!

This software is distributed under the terms of the BSD License.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright notice,
   this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in
   the documentation and/or other materials provided with the
   distribution.

Neither name of CoolServlets.com nor the names of its contributors may
be used to endorse or promote products derived from this software without
specific prior written permission.

THIS SOFTWARE IS PROVIDED BY COOLSERVLETS.COM AND CONTRIBUTORS ``AS IS''
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS
BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY,
OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING INANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
THE POSSIBILITY OF SUCH DAMAGE."

%% The following software may be included in this product: Crimson
v1.1.1 ; Use of any of this software is governed by the terms of the
license below:

/*
 * The Apache Software License, Version 1.1
 *
 *
 * Copyright (c) 1999-2000 The Apache Software Foundation. All rights
 * reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in
 * the documentation and/or other materials provided with the
 * distribution.
 *
 * 3. The end-user documentation included with the redistribution,
 * if any, must include the following acknowledgment:
 * "This product includes software developed by the
 * Apache Software Foundation (http://www.apache.org/)."
 * Alternately, this acknowledgment may appear in the software itself,
 * if and wherever such third-party acknowledgments normally appear.
 *
 * 4. The names "Crimson" and "Apache Software Foundation" must
 * not be used to endorse or promote products derived from this
 * software without prior written permission. For written
 * permission, please contact apache@apache.org.
 *
 * 5. Products derived from this software may not be called "Apache",
 * nor may "Apache" appear in their name, without prior written
 * permission of the Apache Software Foundation.
 *
 * THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
 * WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 * DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
 * LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
 * USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
 * ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
 * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
 * OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 * ====================================================================*
 * This software consists of voluntary contributions made by many
 * individuals on behalf of the Apache Software Foundation and was
 * originally based on software copyright (c) 1999, International
 * Business Machines, Inc., http://www.ibm.com. For more
 * information on the Apache Software Foundation, please see
 * <http://www.apache.org/>.
 */


%% The following software may be included in this product: Xalan J2;
Use of any of this software is governed by the terms of the license below:

                                 Apache License
                           Version 2.0, January 2004
                        http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

   1. Definitions.

      "License" shall mean the terms and conditions for use, reproduction,
      and distribution as defined by Sections 1 through 9 of this
      document.

      "Licensor" shall mean the copyright owner or entity authorized by
      the copyright owner that is granting the License.

      "Legal Entity" shall mean the union of the acting entity and all
      other entities that control, are controlled by, or are under common
      control with that entity. For the purposes of this definition,
      "control" means (i) the power, direct or indirect, to cause the
```

direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work

```
   constitutes direct or contributory patent infringement, then any
   patent licenses granted to You under this License for that Work
   shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
       that You distribute, all copyright, patent, trademark, and
       attribution notices from the Source form of the Work,
       excluding those notices that do not pertain to any part of
       the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its
       distribution, then any Derivative Works that You distribute must
       include a readable copy of the attribution notices contained
       within such NOTICE file, excluding those notices that do not
       pertain to any part of the Derivative Works, in at least one
       of the following places: within a NOTICE text file distributed
       as part of the Derivative Works; within the Source form or
       documentation, if provided along with the Derivative Works; or,
       within a display generated by the Derivative Works, if and
       wherever such third-party notices normally appear. The contents
       of the NOTICE file are for informational purposes only and
       do not modify the License. You may add Your own attribution
       notices within Derivative Works that You distribute, alongside
       or as an addendum to the NOTICE text from the Work, provided
       that such additional attribution notices cannot be construed
       as modifying the License.

   You may add Your own copyright statement to Your modifications and
   may provide additional or different license terms and conditions
   for use, reproduction, or distribution of Your modifications,
   or for any such Derivative Works as a whole, provided Your use,
   reproduction, and distribution of the Work otherwise complies with
   the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
   any Contribution intentionally submitted for inclusion in the Work
   by You to the Licensor shall be under the terms and conditions of
   this License, without any additional terms or conditions.

   Notwithstanding the above, nothing herein shall supersede or modify
   the terms of any separate license agreement you may have executed
   with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
   names, trademarks, service marks, or product names of the Licensor,
   except as required for reasonable and customary use in describing the
   origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
   agreed to in writing, Licensor provides the Work (and each
   Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT
   WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied,
   including, without limitation, any warranties or conditions
   of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
   PARTICULAR PURPOSE. You are solely responsible for determining the
   appropriateness of using or redistributing the Work and assume
   any risks associated with Your exercise of permissions under
```

```
      this License.

   8. Limitation of Liability. In no event and under no legal theory,
      whether in tort (including negligence), contract, or otherwise,
      unless required by applicable law (such as deliberate and grossly
      negligent acts) or agreed to in writing, shall any Contributor be
      liable to You for damages, including any direct, indirect, special,
      incidental, or consequential damages of any character arising as a
      result of this License or out of the use or inability to use the
      Work (including but not limited to damages for loss of goodwill,
      work stoppage, computer failure or malfunction, or any and all
      other commercial damages or losses), even if such Contributor has
      been advised of the possibility of such damages.

   9. Accepting Warranty or Additional Liability. While redistributing
      the Work or Derivative Works thereof, You may choose to offer,
      and charge a fee for, acceptance of support, warranty, indemnity,
      or other liability obligations and/or rights consistent with this
      License. However, in accepting such obligations, You may act only on
      Your own behalf and on Your sole responsibility, not on behalf of
      any other Contributor, and only if You agree to indemnify, defend,
      and hold each Contributor harmless for any liability incurred by,
      or claims asserted against, such Contributor by reason of your
      accepting any such warranty or additional liability.

   END OF TERMS AND CONDITIONS

   APPENDIX: How to apply the Apache License to your work.

      To apply the Apache License to your work, attach the following
      boilerplate notice, with the fields enclosed by brackets "[]"
      replaced with your own identifying information. (Don't include
      the brackets!)  The text should be enclosed in the appropriate
      comment syntax for the file format. We also recommend that a file or
      class name and description of purpose be included on the same
      "printed page" as the copyright notice for easier identification
      within third-party archives.

   Copyright [yyyy] [name of copyright owner]

   Licensed under the Apache License, Version 2.0 (the "License");
   you may not use this file except in compliance with the License.
   You may obtain a copy of the License at

       http://www.apache.org/licenses/LICENSE-2.0

   Unless required by applicable law or agreed to in writing, software
   distributed under the License is distributed on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
   implied. See the License for the specific language governing
   permissions and limitations under the License.



%% The following software may be included in this product: NSIS 1.0j;
Use of any of this software is governed by the terms of the license below:

Copyright (C) 1999-2000 Nullsoft, Inc.

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any damages
arising from the use of this software. Permission is granted to anyone
to use this software for any purpose, including commercial applications,
and to alter it and redistribute it freely, subject to the following
restrictions:

1. The origin of this software must not be misrepresented; you must not
```

claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution. Justin Frankel justin@nullsoft.com"

%% Some Portions licensed from IBM are available at: http://www.ibm.com/software/globalization/icu/

%% Portions Copyright Eastman Kodak Company 1992

%% Lucida is a registered trademark or trademark of Bigelow & Holmes in the U.S. and other countries.

%% Portions licensed from Taligent, Inc.

%% The following software may be included in this product:IAIK PKCS Wrapper; Use of any of this software is governed by the terms of the license below:

Copyright (c) 2002 Graz University of Technology. All rights reserved.

Redistribution and use in source and binary forms, with or without modification,are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:

   "This product includes software developed by IAIK of Graz University of Technology."

   Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. The names "Graz University of Technology" and "IAIK of Graz University of Technology" must not be used to endorse or promote products derived from this software without prior written permission.

5. Products derived from this software may not be called "IAIK PKCS Wrapper", nor may "IAIK" appear in their name, without prior written permission of Graz University of Technology.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE LICENSOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

%% The following software may be included in this product: Document Object Model (DOM) v. Level 3; Use of any of this software is governed by the terms of the license below:

W3C® SOFTWARE NOTICE AND LICENSE

http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231

This work (and included software, documentation such as READMEs, or other related items) is being provided by the copyright holders under the following license. By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications:

1. The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.

2. Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code.

3. Notice of any changes or modifications to the files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.)

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKENO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THEUSE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS,COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL ORCONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission. Title to copyright in this software and any associated documentation will at all times remain with copyright holders.

This formulation of W3C's notice and license became active on December 31 2002. This version removes the copyright ownership notice such that this license can be used with materials other than those owned by the W3C, reflects that ERCIM is now a host of the W3C, includes references to this specific dated version of the license, and removes the ambiguous grant of "use". Otherwise, this version is the same as the previous version and is written so as to preserve the Free Software Foundation's assessment of GPL compatibility and OSI's certification under the Open Source Definition. Please see our Copyright FAQ for common questions about using materials from our site, including specific terms and conditions for packages like libwww, Amaya, and Jigsaw. Other questions about this notice can be directed to site-policy@w3.org.

%% The following software may be included in this product: Xalan, Xerces; Use of any of this software is governed by the terms of the license below:

```
/*
 * The Apache Software License, Version 1.1
 *
 *
 * Copyright (c) 1999-2003 The Apache Software Foundation.  All rights
 * reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in
 *    the documentation and/or other materials provided with the
 *    distribution.
 *
 * 3. The end-user documentation included with the redistribution,
 *    if any, must include the following acknowledgment:
 *       "This product includes software developed by the
 *        Apache Software Foundation (http://www.apache.org/)."
 *    Alternately, this acknowledgment may appear in the software itself,
 *    if and wherever such third-party acknowledgments normally appear.
 *
 * 4. The names "Xerces" and "Apache Software Foundation" must
 *    not be used to endorse or promote products derived from this
 *    software without prior written permission. For written
 *    permission, please contact apache@apache.org.
 *
 * 5. Products derived from this software may not be called "Apache",
 *    nor may "Apache" appear in their name, without prior written
 *    permission of the Apache Software Foundation.
 *
 * THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
 * WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 * DISCLAIMED.  IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
 * LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
 * USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
 * ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
 * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
 * OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 * ====================================================================
 *
 * This software consists of voluntary contributions made by many
 * individuals on behalf of the Apache Software Foundation and was
 * originally based on software copyright (c) 1999, International
 * Business Machines, Inc., http://www.ibm.com.  For more
 * information on the Apache Software Foundation, please see
 *
```

%% The following software may be included in this product: W3C XML Conformance Test Suites v. 20020606; Use of any of this software is governed by the terms of the license below:

W3C® SOFTWARE NOTICE AND LICENSE

Copyright © 1994-2002 World Wide Web Consortium, (Massachusetts Institute ofTechnology, Institut National de Recherche en Informatique et en Automatique,Keio University). All Rights Reserved. http://www.w3.org/Consortium/Legal/

This W3C work (including software, documents, or other related items) is beingprovided by the copyright holders under the following license. By obtaining,using and/or copying this work, you (the licensee) agree that you have read,understood, and will comply with the following terms and conditions:

Permission to use, copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee orroyalty is hereby granted, provided that you include the following on ALL copiesof the software and documentation or portions thereof, including modifications,that you make:

1. The full text of this NOTICE in a location viewable to users of theredistributed or derivative work.

2. Any pre-existing intellectual property disclaimers, notices, or terms andconditions. If none exist, a short notice of the following form (hypertext ispreferred, text is permitted) should be used within the body of any redistributed or derivative code: "Copyright © [$date-of-software] World WideWeb Consortium, (Massachusetts Institute of Technology, Institut National deRecherche en Informatique et en Automatique, Keio University). All RightsReserved. http://www.w3.org/Consortium/Legal/"

3. Notice of any changes or modifications to the W3C files, including thedate changes were made. (We recommend you provide URIs to the location from which the code is derived.)

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKENO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITEDTO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THATTHE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTYPATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL ORCONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION.

The name and trademarks of copyright holders may NOT be used in advertising orpublicity pertaining to the software without specific, written prior permission.Title to copyright in this software and any associated documentation will at alltimes remain with copyright holders.

This formulation of W3C's notice and license became active on August 14 1998 soas to improve compatibility with GPL. This version ensures that W3C softwarelicensing terms are no more restrictive than GPL and consequently W3C softwaremay be distributed in GPL packages. See the older formulation for the policyprior to this date. Please see our Copyright FAQ for common questions aboutusing materials from our site, including specific terms and conditions forpackages like libwww, Amaya, and Jigsaw. Other questions about this notice canbe directed to site-policy@w3.org.

%% The following software may be included in this product: W3C XML Schema Test Collection v. 1.16.2; Use of any of this software is governed by the terms of the license below:

W3C® DOCUMENT NOTICE AND LICENSE

Copyright © 1994-2002 World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique,Keio University).
All Rights Reserved. http://www.w3.org/Consortium/Legal/

Public documents on the W3C site are provided by the copyright holders under thefollowing license. The software or Document Type Definitions (DTDs) associatedwith W3C specifications are governed by the Software Notice. By using and/orcopying this document, or the W3C document from which this statement is linked,you (the licensee) agree that you have read, understood, and will comply withthe following terms and conditions:

Permission to use, copy, and distribute the contents of this document, or theW3C document from which this statement is linked, in any medium for any purpose and without fee or royalty is hereby granted, provided that you include thefollowing on ALL copies of the document, or portions thereof, that you use:

1. A link or URL to the original W3C document.
2. The pre-existing copyright notice of the original author, or if it doesn'texist, a notice of the form: "Copyright © [$date-of-document] World Wide WebConsortium, (Massachusetts Institute of Technology, Institut National deRecherche en Informatique et en Automatique, Keio University). All RightsReserved. http://www.w3.org/Consortium/Legal/" (Hypertext is preferred, but a textual representation is permitted.)
3. If it exists, the STATUS of the W3C document.

When space permits, inclusion of the full text of this NOTICE should be provided. We request that authorship attribution be provided in any software,documents, or other items or products that you create pursuant to the implementation of the contents of this document, or any portion thereof.

No right to create modifications or derivatives of W3C documents is grantedpursuant to this license. However, if additional requirements (documented in the Copyright FAQ) are satisfied, the right to create modifications or derivativesis sometimes granted by the W3C to individuals complying with those requirements.

THIS DOCUMENT IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONSOR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OFMERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE;THAT THE CONTENTS OF THE DOCUMENT ARE SUITABLE FOR ANY PURPOSE; NOR THAT THEIMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS,COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL ORCONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE DOCUMENT OR THE PERFORMANCEOR IMPLEMENTATION OF THE CONTENTS THEREOF.

The name and trademarks of copyright holders may NOT be used in advertising orpublicity pertaining to this document or its contents without specific, writtenprior permission. Title to copyright in this document will at all times remainwith copyright holders.

This formulation of W3C's notice and license became active on April 05 1999 so as to account for the treatment of DTDs, schema's and bindings. See the older formulation for the policy prior to this date. Please see our Copyright FAQ forcommon questions about using materials from our site, including specific terms and conditions for packages like libwww, Amaya, and Jigsaw. Other questions about this notice can be directed to site-policy@w3.org.
webmaster
(last updated by reagle on 1999/04/99.)

%% The following software may be included in this product: Mesa 3-D graphics library v. 5; Use of any of this software is governed by the terms of the license below:

core Mesa code	include/GL/gl.h	Brian Paul	Mesa
GLX driver	include/GL/glx.h	Brian Paul	Mesa
Ext registry	include/GL/glext.h include/GL/glxext.h	SGI	SGI Free B

Mesa license:

The Mesa distribution consists of several components. Different copyrights andlicenses apply to different components. For example, GLUT is copyrighted by MarkKilgard, some demo programs are copyrighted by SGI, some of the Mesa devicedrivers are copyrighted by their authors. See below for a list of Mesa'scomponents and the copyright/license for each.

The core Mesa library is licensed according to the terms of the XFree86copyright (an MIT-style license). This allows integration with the XFree86/DRIproject. Unless otherwise stated, the Mesa source code and documentation islicensed as follows:

Copyright (C) 1999-2003 Brian Paul   All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining acopy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitationthe rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be includedin all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESSOR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL BRIAN PAUL BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

SGI FREE SOFTWARE LICENSE B (Version 1.1 [02/22/2000])
1. Definitions.

1.1 "Additional Notice Provisions" means such additional provisions as appear in the Notice in Original Code under the heading "Additional Notice Provisions."

1.2 "Covered Code" means the Original Code or Modifications, or any combination thereof.

1.3 "Hardware" means any physical device that accepts input, processes input, stores the results of processing, and/or provides output.

1.4 "Larger Work" means a work that combines Covered Code or portions thereof with code not governed by the terms of this License.

1.5 "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.6 "License" means this document.

1.7 "Licensed Patents" means patent claims Licensable by SGI that are infringed by the use or sale of Original Code or any Modifications provided by SGI, or any combination thereof.

1.8 "Modifications" means any addition to or deletion from the substance or structure of the Original Code or any previous Modifications. When Covered Code is released as a series of files, a Modification is: A. Any addition to the contents of a file containing Original Code and/or addition to or deletion from the contents of a file containing previous Modifications.B. Any new file that contains any part of the Original Code or previous Modifications.

1.9 "Notice" means any notice in Original Code or Covered Code, as required by and in compliance with this License.

1.10 "Original Code" means source code of computer software code that is described in the source code Notice required by Exhibit A as Original Code, and updates and error corrections specifically thereto.

1.11 "Recipient" means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License or a future version of this License issued under Section 8. For legal entities, "Recipient" includes any entity that controls, is controlled by, or is under common control with Recipient. For purposes of this definition, "control" of an entity means (a) the power, direct or indirect, to direct or manage such entity, or (b) ownership of fifty percent (50%) or more of the outstanding shares or beneficial ownership of such entity.

1.12 "Recipient Patents" means patent claims Licensable by a Recipient that are infringed by the use or sale of Original Code or any Modifications provided by SGI, or any combination thereof.

1.13 "SGI" means Silicon Graphics, Inc.

1.14 "SGI Patents" means patent claims Licensable by SGI other than the Licensed Patents.

2. License Grant and Restrictions.

2.1 SGI License Grant. Subject to the terms of this License and any third party intellectual property claims, for the duration of intellectual property protections inherent in the Original Code, SGI hereby grants Recipient a worldwide, royalty-free, non-exclusive license, to do the following: (i) under copyrights Licensable by SGI, to reproduce, distribute, create derivative works from, and, to the extent applicable, display and perform the Original Code and/or any Modifications provided by SGI alone and/or as part of a Larger Work; and (ii) under any Licensable Patents, to make, have made, use, sell, offer for sale, import and/or otherwise transfer the Original Code and/or any Modifications provided by SGI. Recipient accepts the terms and conditions of this License by undertaking any of the aforementioned actions. The patent license shall apply to the Covered Code if, at the time any related Modification is added, such addition of the Modification causes such combination to be covered by the Licensed Patents. The patent license in Section 2.1(ii) shall not apply to any other combinations that include the Modification. No patent license is provided under SGI Patents for infringements of SGI Patents by Modifications not provided by SGI or combinations of Original Code and Modifications not provided by SGI.

2.2 Recipient License Grant. Subject to the terms of this License and any third party intellectual property claims, Recipient hereby grants SGI and any other Recipients a worldwide, royalty-free, non-exclusive license, under any Recipient Patents, to make, have made, use, sell, offer for sale, import and/or otherwise transfer the Original Code and/or any Modifications provided by SGI.

2.3 No License For Hardware Implementations. The licenses granted in Section 2.1 and 2.2 are not applicable to implementation in Hardware of the algorithms embodied in the Original Code or any Modifications provided by SGI.

3. Redistributions.

3.1 Retention of Notice/Copy of License. The Notice set forth in Exhibit A, below, must be conspicuously retained or included in any and all redistributions of Covered Code. For distributions of the Covered Code in source code form, the Notice must appear in every file that can include a text comments field; in executable form, the Notice and a copy of this License must appear in related documentation or collateral where the Recipient's rights relating to Covered Code are described. Any Additional Notice Provisions which actually appears in the Original Code must also be retained or included in any and all redistributions of Covered Code.

3.2 Alternative License. Provided that Recipient is in compliance with the terms of this License, Recipient may, so long as without derogation of any of SGI's rights in and to the Original Code, distribute the source code and/or executable version(s) of Covered Code under (1) this License; (2) a license identical to this License but for only such changes as are necessary in order to clarify Recipient's role as licensor of Modifications; and/or (3) a license of Recipient's choosing, containing terms different from this License, provided that the license terms include this Section 3 and Sections 4, 6, 7, 10, 12, and 13, which terms may not be modified or superseded by any other terms of such license. If Recipient elects to use any license other than this License, Recipient must make it absolutely clear that any of its terms which differ from this License are offered by Recipient alone, and not by SGI. It is emphasized that this License is a limited license, and, regardless of the license form employed by Recipient in accordance with this Section 3.2, Recipient may relicense only such rights, in Original Code and Modifications by SGI, as it has actually been granted by SGI in this License.

3.3 Indemnity. Recipient hereby agrees to indemnify SGI for any liability incurred by SGI as a result of any such alternative license terms Recipient offers.4. Termination. This License and the rights granted hereunder will terminate automatically if Recipient breaches any term herein and fails to cure such breach within 30 days thereof. Any sublicense to the Covered Code that is properly granted shall survive any termination of this License, absent termination by the terms of such sublicense. Provisions that, by their nature, must remain in effect beyond the termination of this License, shall survive.

5. No Trademark Or Other Rights.
This License does not grant any rights to: (i) any software apart from the Covered Code, nor shall any other rights or licenses not expressly granted hereunder arise by implication, estoppel or otherwise with respect to the Covered Code; (ii) any trade name, trademark or service mark whatsoever, including without limitation any related right for purposes of endorsement or promotion of products derived from the Covered Code, without prior written permission of SGI; or (iii) any title to or ownership of the Original Code, which shall at all times remains with SGI. All rights in the Original Code not expressly granted under this License are reserved.

6. Compliance with Laws; Non-Infringement.

There are various worldwide laws, regulations, and executive orders applicable to dispositions of Covered Code, including without limitation export, re-export, and import control laws, regulations, and executive orders, of the U.S. government and other countries, and Recipient is reminded it is obliged to obey such laws, regulations, and executive orders. Recipient may not distribute Covered Code that (i) in any way infringes (directly or contributorily) any intellectual property rights of any kind of any other person or entity or (ii) breaches any representation or warranty, express, implied or statutory, to which, under any applicable law, it might be deemed to have been subject.

7. Claims of Infringement.

If Recipient learns of any third party claim that any disposition of Covered Code and/or functionality wholly or partially infringes the third party's intellectual property rights, Recipient will promptly notify SGI of such claim.

8. Versions of the License.

SGI may publish revised and/or new versions of the License from time to time, each with a distinguishing version number. Once Covered Code has been published under a particular version of the License, Recipient may, for the duration of the license, continue to use it under the terms of that version, or choose to use such Covered Code under the terms of any subsequent version published by SGI. Subject to the provisions of Sections 3 and 4 of this License, only SGI may modify the terms applicable to Covered Code created under this License.

9. DISCLAIMER OF WARRANTY.

COVERED CODE IS PROVIDED "AS IS." ALL EXPRESS AND IMPLIED WARRANTIES AND CONDITIONS ARE DISCLAIMED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES AND CONDITIONS OF MERCHANTABILITY, SATISFACTORY QUALITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. SGI ASSUMES NO RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOFTWARE. SHOULD THE SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, SGI ASSUMES NO COST OR LIABILITY FOR SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY IS AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED CODE IS AUTHORIZED HEREUNDER EXCEPT SUBJECT TO THIS DISCLAIMER.

10. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES NOR LEGAL THEORY, WHETHER TORT (INCLUDING, WITHOUT LIMITATION, NEGLIGENCE OR STRICT LIABILITY), CONTRACT, OR OTHERWISE, SHALL SGI OR ANY SGI LICENSOR BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, LOSS OF DATA, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SGI's NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THAT EXCLUSION AND LIMITATION MAY NOT APPLY TO RECIPIENT.

11. Indemnity.

Recipient shall be solely responsible for damages arising, directly or indirectly, out of its utilization of rights under this License. Recipient will defend, indemnify and hold harmless Silicon Graphics, Inc. from and against any loss, liability, damages, costs or expenses (including the payment of reasonable attorneys fees) arising out of Recipient's use, modification, reproduction and distribution of the Covered Code or out of any representation or warranty made by Recipient.

12. U.S. Government End Users.

The Covered Code is a "commercial item" consisting of "commercial computer software" as such terms are defined in title 48 of the Code of Federal Regulations and all U.S. Government End Users acquire only the rights set forth in this License and are subject to the terms of this License.

13. Miscellaneous.

This License represents the complete agreement concerning the its subject matter. If any provision of this License is held to be unenforceable, such provision shall be reformed so as to achieve as nearly as possible the same legal and economic effect as the original provision and the remainder of this License will remain in effect. This License shall be governed by and construed in accordance with the laws of the United States and the State of California as applied to agreements entered into and to be performed entirely within California between California residents. Any litigation relating to this License shall be subject to the exclusive jurisdiction of the Federal Courts of the Northern District of California (or, absent subject matter jurisdiction in such courts, the courts of the State of California), with venue lying exclusively in Santa Clara County, California, with the losing party responsible for costs, including without limitation, court costs and reasonable attorneys fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation that provides that the language of a contract shall be construed against the drafter shall not apply to this License.

Exhibit A

License Applicability. Except to the extent portions of this file are made subject to an alternative license as permitted in the SGI Free Software License B, Version 1.1 (the "License"), the contents of this file are subject only to the provisions of the License. You may not use this file except in compliance with the License. You may obtain a copy of the License at Silicon Graphics, Inc., attn: Legal Services, 1600 Amphitheatre Parkway, Mountain View, CA 94043-1351, or at: http://oss.sgi.com/projects/FreeB

Note that, as provided in the License, the Software is distributed on an "AS IS" basis, with ALL EXPRESS AND IMPLIED WARRANTIES AND CONDITIONS DISCLAIMED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES AND CONDITIONS OF MERCHANTABILITY, SATISFACTORY QUALITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT.Original Code. The Original Code is: [name of software, version number, and release date], developed by Silicon Graphics, Inc. The Original Code is Copyright (c) [dates of first publication, as appearing in the Notice in the Original Code] Silicon Graphics, Inc. Copyright in any portions created by third parties is as indicated elsewhere herein. All Rights Reserved.Additional Notice Provisions: [such additional provisions, if any, as appear in the Notice in the Original Code under the heading "Additional Notice Provisions"]

%% The following software may be included in this product: Byte Code Engineering Library (BCEL) v. 5; Use of any of this software is governed by the terms of the license below:

Apache Software License

```
/* ====================================================================
 * The Apache Software License, Version 1.1
 *
 * Copyright (c) 2001 The Apache Software Foundation.  Allrights
 * reserved.
 *
 * Redistribution and use in source and binary forms, withor without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in
 *    the documentation and/or other materials provided with the
 *    distribution.
 *
 * 3. The end-user documentation included with the redistribution,
 *    if any, must include the following acknowledgment:
 *       "This product includes software developed by the
 *        Apache Software Foundation (http://www.apache.org/)."
 *    Alternately, this acknowledgment may appear in the software itself,
 *    if and wherever such third-party acknowledgments normally appear.
 *
 * 4. The names "Apache" and "Apache Software Foundation"and
 *    "Apache BCEL" must not be used to endorse or promote products
 *    derived from this software without prior written permission. For
 *    written permission, please contact apache@apache.org.                *
 * 5. Products derived from this software may not be called "Apache",
 *    "Apache BCEL", nor may "Apache" appear in their name,without
 *    prior written permission of the Apache Software Foundation.
 *
 * THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
 * WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 * DISCLAIMED.  IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
 * LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
```

%% The following software may be included in this product: Regexp, Regular Expression Package v. 1.2; Use of any of this software is governed by the terms of the license below:

%% The following software may be included in this product: CUP Parser Generator for Java v. 0.10k; Use of any of this software is governed by the terms of the license below:

CUP Parser Generator Copyright Notice, License, and Disclaimer

Copyright 1996-1999 by Scott Hudson, Frank Flannery, C. Scott Ananian

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided thatthe above copyright notice appear in all copies and that both the copyright notice and this permission notice and warranty disclaimer appear in supporting documentation, and that the names of the authors or their employersnot be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

The authors and their employers disclaim all warranties with regard to this software, including all implied warranties of merchantability and fitness. In no event shall the authors or their employers be liable for anyspecial, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of this software.

%% The following software may be included in this product: JLex: A Lexical Analyzer Generator for Java v. 1.2.5; Use of any of this software is governed by the terms of the license below:

JLEX COPYRIGHT NOTICE, LICENSE AND DISCLAIMER.

Copyright 1996-2003 by Elliot Joel Berk and C. Scott Ananian

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright noticeappear in all copies and that both the copyright notice and this permission notice and warrantydisclaimer appear in supporting documentation, and that the name of the authors or their employersnot be used in advertising or publicity pertaining to distribution of the software withoutspecific, written prior permission.

The authors and their employers disclaim all warranties with regard to thissoftware, including all implied warranties of merchantability and fitness. In no event shall the authorsor their employers be liable for any special, indirect or consequential damages or any damageswhatsoever resulting from loss of use, data or profits, whether in an action of contract, negligenceor other tortious action, arising out of or in connection with the use or performance of thissoftware.

Java is a trademark of Sun Microsystems, Inc. References to the Java programminglanguage in relation to JLex are not meant to imply that Sun endorses this product.

%% The following software may be included in this product: SAX v. 2.0.1; Use of any of this software is governed by the terms of the license below:

Copyright Status

SAX is free!

In fact, it's not possible to own a license to SAX, since it's been placed in the public domain.

No Warranty

Because SAX is released to the public domain, there is no warranty for the design or for the software implementation, to the extent permitted by applicable law. Except when otherwise stated in writing the copyright holders and/or other parties provide SAX "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of SAX is with you. Should SAX prove defective, you assume the cost of all necessary servicing, repair or correction.

In no event unless required by applicable law or agreed to in writing will any copyright holder, or any other party who may modify and/or redistribute SAX, be liable to you for damages, including any general, special, incidental or consequential damages arising out of the use or inability to use SAX (including but not limited to loss of data or data being rendered inaccurate or losses sustained by you or third parties or a failure of the SAX to operate with any other programs), even if such holder or other party has been advised of the possibility of such damages.

Copyright Disclaimers

This page includes statements to that effect by David Megginson, who would have been able to claim copyright for the original work. SAX 1.0

Version 1.0 of the Simple API for XML (SAX), created collectively by the membership of the XML-DEV mailing list, is hereby released into the public domain.

No one owns SAX: you may use it freely in both commercial and non-commercial applications, bundle it with your software distribution, include it on a CD-ROM, list the source code in a book, mirror the documentation at your own web site, or use it in any other way you see fit.

David Megginson, sax@megginson.com
1998-05-11

SAX 2.0

I hereby abandon any property rights to SAX 2.0 (the Simple API for XML), and release all of the SAX 2.0 source code, compiled code, and documentation contained in this distribution into the Public Domain. SAX comes with NO WARRANTY or guarantee of fitness for any purpose.

David Megginson, david@megginson.com
2000-05-05

%% The following software may be included in this product: Cryptix;
Use of any of this software is governed by the terms of the license below:

Cryptix General License

Copyright © 1995-2003 The Cryptix Foundation Limited. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE CRYPTIX FOUNDATION LIMITED AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS ORIMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FORA PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE CRYPTIX FOUNDATION LIMITED OR CONTRIBUTORS BELIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOTLIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESSINTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OFTHE POSSIBILITY OF SUCH DAMAGE.

%% The following software may be included in this product: W3C XML Schema Test Collection; Use of any of this software is governed by the terms of the license below:

W3C® DOCUMENT NOTICE AND LICENSE

Copyright © 1994-2002 World Wide Web Consortium, (Massachusetts Institute ofTechnology, Institut National de Recherche en Informatique et en Automatique,Keio University). All Rights Reserved. http://www.w3.org/Consortium/Legal/

Public documents on the W3C site are provided by the copyright holders under thefollowing license. The software or Document Type Definitions (DTDs) associatedwith W3C specifications are governed by the Software Notice. By using and/orcopying this document, or the W3C document from which this statement is linked,you (the licensee) agree that you have read, understood, and will comply withthe following terms and conditions:

Permission to use, copy, and distribute the contents of this document, or theW3C document from which this statement is linked, in any medium for any purposeand without fee or royalty is hereby granted, provided that you include thefollowing on ALL copies of the document, or portions thereof, that you use:

1. A link or URL to the original W3C document.
2. The pre-existing copyright notice of the original author, or if it doesn'texist, a notice of the form: "Copyright © [$date-of-document] World Wide WebConsortium, (Massachusetts Institute of Technology, Institut National deRecherche en Informatique et en Automatique, Keio University). All RightsReserved. http://www.w3.org/Consortium/Legal/" (Hypertext is preferred, but atextual representation is permitted.)

3. If it exists, the STATUS of the W3C document.

When space permits, inclusion of the full text of this NOTICE should beprovided. We request that authorship attribution be provided in any software,documents, or other items or products that you create pursuant to the implementation of the contents of this document, or any portion thereof. No right to create modifications or derivatives of W3C documents is grantedpursuant to this license. However, if additional requirements (documented in theCopyright FAQ) are satisfied, the right to create modifications or derivativesis sometimes granted by the W3C to individuals complying with those requirements.

THIS DOCUMENT IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONSOR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OFMERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE;THAT THE CONTENTS OF THE DOCUMENT ARE SUITABLE FOR ANY PURPOSE; NOR THAT THEIMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS,COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL ORCONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE DOCUMENT OR THE PERFORMANCEOR IMPLEMENTATION OF THE CONTENTS THEREOF.

The name and trademarks of copyright holders may NOT be used in advertising orpublicity pertaining to this document or its contents without specific, writtenprior permission. Title to copyright in this document will at all times remainwith copyright holders.

This formulation of W3C's notice and license became active on April 05 1999 soas to account for the treatment of DTDs, schema's and bindings. See the olderformulation for the policy prior to this date. Please see our Copyright FAQ forcommon questions about using materials from our site, including specific termsand conditions for packages like libwww, Amaya, and Jigsaw. Other questionsabout this notice can be directed to site-policy@w3.org.
webmaster
(last updated by reagle on 1999/04/99.)

%% The following software may be included in this product: Stax API;
Use of any of this software is governed by the terms of the license below:

Streaming API for XML (JSR-173) Specification
Reference Implementation
License Agreement

READ THE TERMS OF THIS (THE "AGREEMENT") CAREFULLY BEFORE VIEWING OR USING THESOFTWARE LICENS ED HEREUNDER. BY VIEWING OR USING THE SOFTWARE, YOU AGREE TO THE TERMS OF THISAGREEMENT. IF YOU ARE ACCESSING THE SOFTWARE ELECTRONICALLY, INDICATE YOUR ACCEPTANCE OF THESETERMS BY SELE CTING THE "ACCEPT" BUTTON AT THE END OF THIS AGREEMENT. IF YOU DO NOT AGREE TOALL THESE TERMS , PROMPTLY RETURN THE UNUSED SOFTWARE TO ORIGINAL CONTRIBUTOR, DEFINED HEREIN.

1.0 DEFINITIONS.

1.1. "BEA" means BEA Systems, Inc., the licensor of the Original Code.

1.2. "Contributor" means BEA and each entity that creates or contributes to the creation of Modifications.

1.3. "Covered Code" means the Original Code or Modifications or the combination of the Original Code and Modifications, in each case including portions thereof and corresponding documentation released with the source code.

1.4. "Executable" means Covered Code in any form other than Source Code.

1.5. "FCS" means first commercial shipment of a product.

1.6. "Modifications" means any addition to or deletion from the substance or structure of either the Original Code or any previous Modifications. When Covered Code isreleased as a series of files, a Modification is:

(a) Any addition to or deletion from the contents of a file containing OriginalCode or previous Modifications.

(b) Any new file that contains any part of the Original Code or previous Modifications.

1.7. "Original Code" means Source Code of computer software code Reference Implementation.

1.8. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent for which the grantor has the right to grant a license.

1.9. "Reference Implementation" means the prototype or "proof of concept" implementation of the Specification developed and made available for license by or on behalf of BEA.

1.10. "Source Code" means the preferred form of the Covered Code for making modifications to it, including all modules it contains, plus any associated documentation,interface definition files, scripts used to control compilation and installation of an Executable, or source code differential comparisons against either the Original Code or another well known, available Covered Code of the Contributor's choice.

1.11. "Specification" means the written specification for the Streaming API for XML, Java technology developed pursuant to the Java Community Process.

1.12. "Technology Compatibility Kit" or "TCK" means the documentation, testing tools and test suites associated with the Specification as may be revised by BEA from time to time, that is provided so that an implementer of the Specification may determine if its implementation is compliant with the Specification.

1.13. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this Agreement or a future version of this Agreement issued under Section 6.1. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of suchentity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of theoutstanding shares or beneficial ownership of such entity.

2.0 SOURCE CODE LICENSE.

2.1. Copyright Grant.
Subject to the terms of this Agreement, each Contributor hereby grants You a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Covered Code of such Contributor, if any, and such derivative works, in Source Code and Executable form.

2.2. Patent Grant.
Subject to the terms of this Agreement, each Contributor hereby grants You a non-exclusive, worldwide, royalty-free patent license under the Patent Claims to make, use, sell, offer to sell, import and otherwise transfer the Covered Code prepared and provided by such Contributor, if any, in Source Code and Executable form. This patent license shall apply to the Covered Code if, at the time a Modification is added by the Contributor, such addition of the Modification causes such combination to be covered by the Patent Claims. The patent license shall not apply to any other combinations which include the Modification.

2.3. Conditions to Grants.
You understand that although each Contributor grants the licenses to the Covered Code prepared by it, no assurances are provided by any Contributor that the Covered Code does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to You for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, You hereby assume sole responsibility to secure any other intellectual property rights needed, if any. For example, if a thirdparty patent license is required to allow You to distribute Covered Code, it is Your responsibility to acquire that license before distributing such code.

2.4. Contributors' Representation.
Each Contributor represents that to its knowledge it has sufficient copyright rights in the Covered Code it provides, if any, to grant the copyright license set forth in this Agreement.

3.0 DISTRIBUION RESTRICTIONS.

3.1. Application of Agreement.

The Modifications which You create or to which You contribute are governed by the terms of this Agreement, including without limitation

Section 2.0. The Source Code version of Covered Code may be distributed only under the terms of this Agreement or a future version of this Agreement released under Section 6.1, and You must include a copy of this Agreement with every copy of the Source Code You distribute. You may not offer or impose any terms on any Source Code version that alters or restricts the applicable version of this Agreement or therecipients' rights hereunder. However, You may include an additional document offering theadditional rights d escribed in Section 3.3.

3.2. Description of Modifications.

You must cause all Covered Code to which You contribute to contain a file documenting the changes You made to create that Covered Code and the date of any change. You must include a prominent statement that the Modification is derived, directly or indirectly, from Original Code provided by BEA and including the name of BEA in (a) the Source Code, and (b) in any notice in an Executable version or related documentation in which You describe the origin or ownership of the Covered Code.

%% The following software may be included in this product: X Window System; Use of any of this software is governed by the terms of the license below:

Copyright The Open Group

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation.

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESSFOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE OPEN GROUP BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of The Open Group shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from The Open Group.

Portions also covered by other licenses as noted in the above URL.

%% The following software may be included in this product: dom4j v. 1.6; Use of any of this software is governed by the terms of the license below:

Redistribution and use of this software and associated documentation ("Software"), with or without modification, are permitted provided that thefollowing conditions are met:

1. Redistributions of source code must retain copyright statements and notices. Redistributions must also contain a copy of this document.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name "DOM4J" must not be used to endorse or promote products derived from this Software without prior written permission of MetaStuff, Ltd. For written permission, please contact dom4j-info@metastuff.com.
4. Products derived from this Software may not be called "DOM4J" nor may "DOM4J" appear in their names without prior written permission of MetaStuff,Ltd. DOM4J is a registered trademark of MetaStuff, Ltd.

```
   this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

```
%% The following software may be included in this product: libpng official
PNG reference library; Use of any of this software is governed by the
terms of the license below:

This copy of the libpng notices is provided for your convenience.  In case
of any discrepancy between this copy and the notices in the file png.h
that is included in the libpng distribution, the latter shall prevail.

COPYRIGHT NOTICE, DISCLAIMER, and LICENSE:

If you modify libpng you may insert additional notices immediately
following this sentence.

libpng version 1.2.6, December 3, 2004, is Copyright (c) 2004 Glenn
Randers-Pehrson, and is distributed according to the same disclaimer and
license as libpng-1.2.5 with the following individual added to the list
of Contributing Authors
   Cosmin Truta

libpng versions 1.0.7, July 1, 2000, through 1.2.5 - October 3, 2002,
are Copyright (c) 2000-2002 Glenn Randers-Pehrson, and are distributed
according to the same disclaimer and license as libpng-1.0.6with the
following individuals added to the list of Contributing Authors
   Simon-Pierre Cadieux
   Eric S. Raymond
   Gilles Vollant

and with the following additions to the disclaimer:

   There is no warranty against interference with your enjoyment of
   the library or against infringement. There is no warranty that our
   efforts or the library will fulfill any of your particular purposes
   or needs. This library is provided with all faults, and the entire
   risk of satisfactory quality, performance, accuracy, and effort is
   with the user.

libpng versions 0.97, January 1998, through 1.0.6, March 20, 2000, are
Copyright (c) 1998, 1999 Glenn Randers-Pehrson, and are distributed
according to the same disclaimer and license as libpng-0.96,with the
following individuals added to the list of Contributing Authors:
   Tom Lane
   Glenn Randers-Pehrson
   Willem van Schaik

libpng versions 0.89, June 1996, through 0.96, May 1997, are Copyright (c)
1996, 1997 Andreas Dilger Distributed according to the same disclaimer
and license as libpng-0.88, with the following individuals added to the
list of Contributing Authors:
   John Bowler
   Kevin Bracey
   Sam Bushell
   Magnus Holmgren
   Greg Roelofs
   Tom Tanner

libpng versions 0.5, May 1995, through 0.88, January 1996, are
Copyright (c) 1995, 1996 Guy Eric Schalnat, Group 42, Inc.

For the purposes of this copyright and license, "Contributing Authors"is
defined as the following set of individuals:
   Andreas Dilger
   Dave Martindale
   Guy Eric Schalnat
   Paul Schmidt
   Tim Wegner

The PNG Reference Library is supplied "AS IS".  The Contributing
Authorsand Group 42, Inc. disclaim all warranties, expressed or implied,
including, without limitation, the warranties of merchantability and
offitness for any purpose. The Contributing Authors and Group 42,
Inc. assume no liability for direct, indirect, incidental, special,
exemplary,or consequential damages, which may result from the use of the
PNG Reference Library, even if advised of the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute this
source code, or portions hereof, for any purpose, without fee, subject
to the following restrictions:

1. The origin of this source code must not be misrepresented.

2. Altered versions must be plainly marked as such and must not
   be misrepresented as being the original source.

3. This Copyright notice may not be removed or altered from any
   source or altered source distribution.

The Contributing Authors and Group 42, Inc. specifically permit,
without fee, and encourage the use of this source code as a component
to supporting the PNG file format in commercial products. If you use
thissource code in a product, acknowledgment is not required but would
be appreciated.

A "png_get_copyright" function is available, for convenient use in
"about" boxes and the like:

   printf("%s",png_get_copyright(NULL));

Also, the PNG logo (in PNG format, of course) is supplied in the
files "pngbar.png" and "pngbar.jpg (88x31) and "pngnow.png" (98x31).

Libpng is OSI Certified Open Source Software. OSI Certified Open Source
is a certification mark of the Open Source Initiative.

Glenn Randers-Pehrson
glennrp at users.sourceforge.net
December 3, 2004

%% The following software may be included in this product: Libungif -
An uncompressed GIF library; Use of any of this software is governed by
the terms of the license below:

The GIFLIB distribution is Copyright (c) 1997  Eric S. Raymond

Permission is hereby granted, free of charge, to any person obtaining a
copyof this software and associated documentation files (the "Software"),
to deal in the Software without restriction, including without limitation
the rights to use, copy, modify, merge, publish, distribute, sublicense,
and/or sell copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included
```

```
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.  IN NO EVENT SHALL
THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR
OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.


%% The following software may be included in this product: Ant; Use of
any of this software is governed by the terms of the license below:

License
The Apache Software License Version 2.0

The Apache Software License Version 2.0 applies to all releases of Ant
startingwith ant 1.6.1

/*
 *                                 Apache License
 *                           Version 2.0, January 2004
 *                        http://www.apache.org/licenses/
 *
 *   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
 *
 *   1. Definitions.
 *
 *      "License" shall mean the terms and conditions for use, reproduction,
 *      and distribution as defined by Sections 1 through 9 of this document.
 *
 *      "Licensor" shall mean the copyright owner or entity authorized by
 *      the copyright owner that is granting the License.
 *
 *      "Legal Entity" shall mean the union of the acting entity and all
 *      other entities that control, are controlled by, or are under common
 *      control with that entity. For the purposes of this definition,
 *      "control" means (i) the power, direct or indirect, to cause the
 *      direction or management of such entity, whether by contract or
 *      otherwise, or (ii) ownership of fifty percent (50%) or more of the
 *      outstanding shares, or (iii) beneficial ownership of such entity.
 *
 *      "You" (or "Your") shall mean an individual or Legal Entity
 *      exercising permissions granted by this License.
 *
 *      "Source" form shall mean the preferred form for making modifications,
 *      including but not limited to software source code, documentation
 *      source, and configuration files.
 *
 *      "Object" form shall mean any form resulting from mechanical
 *      transformation or translation of a Source form, including but
 *      not limited to compiled object code, generated documentation,
 *      and conversions to other media types.
 *
 *      "Work" shall mean the work of authorship, whether in Source or
 *      Object form, made available under the License, as indicated by a
 *      copyright notice that is included in or attached to the work
 *      (an example is provided in the Appendix below).
 *
 *      "Derivative Works" shall mean any work, whether in Source or Object
 *      form, that is based on (or derived from) the Work and for which the
 *      editorial revisions, annotations, elaborations, or other modifications
 *      represent, as a whole, an original work of authorship. For the purposes
 *      of this License, Derivative Works shall not include works that remain
 *      separable from, or merely link (or bind by name) to the interfaces of,
 *      the Work and Derivative Works thereof.
 *
```

```
*      "Contribution" shall mean any work of authorship, including
*      the original version of the Work and any modifications or additions
*      to that Work or Derivative Works thereof, that is intentionally
*      submitted to Licensor for inclusion in the Work by the copyright owner
*      or by an individual or Legal Entity authorized to submit on behalf of
*      the copyright owner. For the purposes of this definition, "submitted"
*      means any form of electronic, verbal, or written communication sent
*      to the Licensor or its representatives, including but not limited to
*      communication on electronic mailing lists, source code control systems,
*      and issue tracking systems that are managed by, or on behalf of, the
*      Licensor for the purpose of discussing and improving the Work, but
*      excluding communication that is conspicuously marked or otherwise
*      designated in writing by the copyright owner as "Not a Contribution."
*
*      "Contributor" shall mean Licensor and any individual or Legal Entity
*      on behalf of whom a Contribution has been received by Licensor and
*      subsequently incorporated within the Work.
*
*   2. Grant of Copyright License. Subject to the terms and conditions of
*      this License, each Contributor hereby grants to You a perpetual,
*      worldwide, non-exclusive, no-charge, royalty-free, irrevocable
*      copyright license to reproduce, prepare Derivative Works of,
*      publicly display, publicly perform, sublicense, and distribute the
*      Work and such Derivative Works in Source or Object form.
*
*   3. Grant of Patent License. Subject to the terms and conditions of
*      this License, each Contributor hereby grants to You a perpetual,
*      worldwide, non-exclusive, no-charge, royalty-free, irrevocable
*      (except as stated in this section) patent license to make, have made,
*      use, offer to sell, sell, import, and otherwise transfer the Work,
*      where such license applies only to those patent claims licensable
*      by such Contributor that are necessarily infringed by their
*      Contribution(s) alone or by combination of their Contribution(s)
*      with the Work to which such Contribution(s) was submitted. If You
*      institute patent litigation against any entity (including a
*      cross-claim or counterclaim in a lawsuit) alleging that the Work
*      or a Contribution incorporated within the Work constitutes direct
*      or contributory patent infringement, then any patent licenses
*      granted to You under this License for that Work shall terminate
*      as of the date such litigation is filed.
*
*   4. Redistribution. You may reproduce and distribute copies of the
*      Work or Derivative Works thereof in any medium, with or without
*      modifications, and in Source or Object form, provided that You
*      meet the following conditions:
*
*      (a) You must give any other recipients of the Work or
*          Derivative Works a copy of this License; and
*
*      (b) You must cause any modified files to carry prominent notices
*          stating that You changed the files; and
*
*      (c) You must retain, in the Source form of any Derivative Works
*          that You distribute, all copyright, patent, trademark, and
*          attribution notices from the Source form of the Work,
*          excluding those notices that do not pertain to any part of
*          the Derivative Works; and
*
*      (d) If the Work includes a "NOTICE" text file as part of its
*          distribution, then any Derivative Works that You distribute must
*          include a readable copy of the attribution notices contained
*          within such NOTICE file, excluding those notices that do not
*          pertain to any part of the Derivative Works, in at least one
*          of the following places: within a NOTICE text file distributed
*          as part of the Derivative Works; within the Source form or
*          documentation, if provided along with the Derivative Works; or,
*          within a display generated by the Derivative Works, if and
*          wherever such third-party notices normally appear. The contents
*          of the NOTICE file are for informational purposes only and
*          do not modify the License. You may add Your own attribution
*          notices within Derivative Works that You distribute, alongside
*          or as an addendum to the NOTICE text from the Work, provided
*          that such additional attribution notices cannot be construed
*          as modifying the License.
*
*      You may add Your own copyright statement to Your modifications and
*      may provide additional or different license terms and conditions
*      for use, reproduction, or distribution of Your modifications, or
*      for any such Derivative Works as a whole, provided Your use,
*      reproduction, and distribution of the Work otherwise complies with
*      the conditions stated in this License.
*
*   5. Submission of Contributions. Unless You explicitly state otherwise,
*      any Contribution intentionally submitted for inclusion in the Work
*      by You to the Licensor shall be under the terms and conditions of
*      this License, without any additional terms or conditions.
*      Notwithstanding the above, nothing herein shall supersede or modify
*      the terms of any separate license agreement you may have executed
*      with Licensor regarding such Contributions.
*
*   6. Trademarks. This License does not grant permission to use the trade
*      names, trademarks, service marks, or product names of the Licensor,
*      except as required for reasonable and customary use in describing the
*      origin of the Work and reproducing the content of the NOTICE file.
*
*   7. Disclaimer of Warranty. Unless required by applicable law or
*      agreed to in writing, Licensor provides the Work (and each
*      Contributor provides its Contributions) on an "AS IS" BASIS,
*      WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
*      implied, including, without limitation, any warranties or conditions
*      of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
*      PARTICULAR PURPOSE. You are solely responsible for determining the
*      appropriateness of using or redistributing the Work and assume any
*      risks associated with Your exercise of permissions under this License.
*
*   8. Limitation of Liability. In no event and under no legal theory,
*      whether in tort (including negligence), contract, or otherwise,
*      unless required by applicable law (such as deliberate and grossly
*      negligent acts) or agreed to in writing, shall any Contributor be
*      liable to You for damages, including any direct, indirect, special,
*      incidental, or consequential damages of any character arising as a
*      result of this License or out of the use or inability to use the
*      Work (including but not limited to damages for loss of goodwill,
*      work stoppage, computer failure or malfunction, or any and all
*      other commercial damages or losses), even if such Contributor
*      has been advised of the possibility of such damages.
*
*   9. Accepting Warranty or Additional Liability. While redistributing
*      the Work or Derivative Works thereof, You may choose to offer,
*      and charge a fee for, acceptance of support, warranty, indemnity,
*      or other liability obligations and/or rights consistent with this
*      License. However, in accepting such obligations, You may act only
*      on Your own behalf and on Your sole responsibility, not on behalf
*      of any other Contributor, and only if You agree to indemnify,
*      defend, and hold each Contributor harmless for any liability
*      incurred by, or claims asserted against, such Contributor by reason
*      of your accepting any such warranty or additional liability.
*
*   END OF TERMS AND CONDITIONS
*
*   APPENDIX: How to apply the Apache License to your work.
*
*      To apply the Apache License to your work, attach the following
*      boilerplate notice, with the fields enclosed by brackets "[]"
*      replaced with your own identifying information. (Don't include
*      the brackets!)  The text should be enclosed in the appropriate
```

```
 *      comment syntax for the file format. We also recommend that a
 *      file or class name and description of purpose be included on the
 *      same "printed page" as the copyright notice for easier
 *      identification within third-party archives.
 *
 *   Copyright [yyyy] Apache Software Foundation
 *
 *   Licensed under the Apache License, Version 2.0 (the "License");
 *   you may not use this file except in compliance with the License.
 *   You may obtain a copy of the License at
 *
 *       http://www.apache.org/licenses/LICENSE-2.0
 *
 *   Unless required by applicable law or agreed to in writing, software
 *   distributed under the License is distributed on an "AS IS" BASIS,
 *   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 *   See the License for the specific language governing permissions and
 *   limitations under the License.
 */
```

You can download the original license file here.

The License is accompanied by a NOTICE

```
   =========================================================================
   ==  NOTICE file corresponding to the section 4 d of                    ==
   ==  the Apache License, Version 2.0,                                   ==
   ==  in this case for the Apache Ant distribution.                      ==
   =========================================================================
   This product includes software developed by
   The Apache Software Foundation (http://www.apache.org/).

   This product includes also software developed by :
     - the W3C consortium (http://www.w3c.org),
     - the SAX project (http://www.saxproject.org)

   Please read the different LICENSE files present in the root directory of
   this distribution.

   The names "Ant" and  "Apache Software Foundation"  must not be used to
   endorse  or promote  products derived  from this  software without prior
   written permission. For written permission, please contact
   apache@apache.org.
```

The Apache Software License, Version 1.1

The Apache Software License, Version 1.1, applies to all versions of up to ant1.6.0 included.

```
/*
 * ============================================================================
 *                   The Apache Software License, Version 1.1
 * ============================================================================
 *
 *    Copyright (C) 2000-2003 The Apache Software Foundation. All
 *    rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without modifica-
 * tion, are permitted provided that the following conditions are met:
 *
 * 1. Redistributions of  source code must  retain the above copyright  notice,
 *    this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright notice,
 *    this list of conditions and the following disclaimer in the documentation
 *    and/or other materials provided with the distribution.
 *
 * 3. The end-user documentation included with the redistribution, if any, musti
 *    include  the following  acknowledgment:  "This product includes  softwarei
 *    developed  by the  Apache Software Foundation  (http://www.apache.org/)."
 *    Alternately, this  acknowledgment may  appear in the software itself,  if
 *    and wherever such third-party acknowledgments normally appear.
 *
 * 4. The names "Ant" and  "Apache Software Foundation"  must not be used to
 *    endorse  or promote  products derived  from this  software without  prior
 *    written permission. For written permission, please contact
 *    apache@apache.org.
 *
 * 5. Products  derived from this software may not  be called "Apache", nor may
 *    "Apache" appear  in their name,  without prior written permission  of the
 *    Apache Software Foundation.
 *
 * THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES,
 * INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND
 * FITNESS  FOR A PARTICULAR  PURPOSE ARE  DISCLAIMED.  IN NO  EVENT SHALL  THE
 * APACHE SOFTWARE  FOUNDATION  OR ITS CONTRIBUTORS  BE LIABLE FOR  ANY DIRECT,
 * INDIRECT, INCIDENTAL, SPECIAL,  EXEMPLARY, OR CONSEQUENTIAL  DAMAGES (INCLU-
 * DING, BUT NOT LIMITED TO, PROCUREMENT  OF SUBSTITUTE GOODS OR SERVICES; LOSS
 * OF USE, DATA, OR  PROFITS; OR BUSINESS  INTERRUPTION)  HOWEVER CAUSED AND ON
 * ANY  THEORY OF LIABILITY,  WHETHER  IN CONTRACT,  STRICT LIABILITY,  OR TORT
 * (INCLUDING  NEGLIGENCE OR  OTHERWISE) ARISING IN  ANY WAY OUT OF THE  USE OF
 * THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 *
 * This software  consists of voluntary contributions made  by many individuals
 * on behalf of the  Apache Software Foundation.  For more  information  on the
 * Apache Software Foundation, please see .
 *
 */
```

%% The following software may be included in this product: XML Resolver library; Use of any of this software is governed by the terms of the license below:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical

```
   transformation or translation of a Source form, including but
   not limited to compiled object code, generated documentation,
   and conversions to other media types.

   "Work" shall mean the work of authorship, whether in Source or
   Object form, made available under the License, as indicated by a
   copyright notice that is included in or attached to the work
   (an example is provided in the Appendix below).

   "Derivative Works" shall mean any work, whether in Source or
   Object form, that is based on (or derived from) the Work and
   for which the editorial revisions, annotations, elaborations,
   or other modifications represent, as a whole, an original work
   of authorship. For the purposes of this License, Derivative Works
   shall not include works that remain separable from, or merely link
   (or bind by name) to the interfaces of, the Work and Derivative
   Works thereof.

   "Contribution" shall mean any work of authorship, including the
   original version of the Work and any modifications or additions
   to that Work or Derivative Works thereof, that is intentionally
   submitted to Licensor for inclusion in the Work by the copyright
   owner or by an individual or Legal Entity authorized to submit
   on behalf of the copyright owner. For the purposes of this
   definition, "submitted" means any form of electronic, verbal, or
   written communication sent to the Licensor or its representatives,
   including but not limited to communication on electronic mailing
   lists, source code control systems, and issue tracking systems
   that are managed by, or on behalf of, the Licensor for the purpose
   of discussing and improving the Work, but excluding communication
   that is conspicuously marked or otherwise designated in writing
   by the copyright owner as "Not a Contribution."

   "Contributor" shall mean Licensor and any individual or Legal
   Entity on behalf of whom a Contribution has been received by
   Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute
   the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have
   made, use, offer to sell, sell, import, and otherwise transfer
   the Work, where such license applies only to those patent claims
   licensable by such Contributor that are necessarily infringed
   by their Contribution(s) alone or by combination of their
   Contribution(s) with the Work to which such Contribution(s) was
   submitted. If You institute patent litigation against any entity
   (including a cross-claim or counterclaim in a lawsuit) alleging
   that the Work or a Contribution incorporated within the Work
   constitutes direct or contributory patent infringement, then any
   patent licenses granted to You under this License for that Work
   shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and
```

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has

been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

%% The following software may be included in this product: ICU4J; Use of any of this software is governed by the terms of the license below:

ICU License - ICU 1.8.1 and later
COPYRIGHT AND PERMISSION NOTICE
Copyright (c) 1995-2003 International Business Machines Corporation and others
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaininga copy of this software and associated documentation files (the "Software"), todeal in the Software without restriction, including without limitation therights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so,provided that the above copyright notice(s) and this permission notice appear inall copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation. THE SOFTWARE IS PROVIDED"AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOTLIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

%% The following software may be included in this product: NekoHTML; Use of any of this software is governed by the terms of the license below:

The CyberNeko Software License, Version 1.0

(C) Copyright 2002,2003, Andy Clark. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by Andy Clark." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. The names "CyberNeko" and "NekoHTML" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact andy@cyberneko.net.

5. Products derived from this software may not be called "CyberNeko", nor may "CyberNeko" appear in their name, without prior written permission of the author.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR OTHER CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

==
This license is based on the Apache Software License, version 1.1

%% The following software may be included in this product: Jing; Use of any of this software is governed by the terms of the license below:

Jing Copying Conditions

Copyright (c) 2001-2003 Thai Open Source Software Center Ltd
All rights reserved.

Redistribution and use in source and binary forms, with or without modification,are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright
  notice,this list of conditions and the following disclaimer in the
  documentation and/or other materials provided with the distribution.

* Neither the name of the Thai Open Source Software Center Ltd nor the
  names of its contributors may be used to endorse or promote products
  derived from thissoftware without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
IS" ANDANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
THE IMPLIEDWARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR

PURPOSE AREDISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS
BE LIABLE FOR ANYDIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY,
OR CONSEQUENTIAL DAMAGES(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES;LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ONANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT(INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THISSOFTWARE, EVEN IF ADVISED OF
THE POSSIBILITY OF SUCH DAMAGE.

%% The following software may be included in this product: RelaxNGCC;
Use of any of this software is governed by the terms of the license below:

Copyright (c) 2000-2003 Daisuke Okajima and Kohsuke Kawaguchi.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

3. The end-user documentation included with the redistribution, if
any, must include the following acknowledgment:

> "This product includes software developed by Daisuke Okajima
> and Kohsuke Kawaguchi (http://relaxngcc.sf.net/)."

Alternately, this acknowledgment may appear in the software itself,
if and wherever such third-party acknowledgments normally appear.

4. The names of the copyright holders must not be used to endorse or
promote products derived from this software without prior written
permission. For written permission, please contact the copyright
holders.

5. Products derived from this software may not be called "RELAXNGCC",
nor may "RELAXNGCC" appear in their name, without prior written
permission of the copyright holders.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES,
INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY
AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.IN NO EVENT SHALL
THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING
IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.

%% The following software may be included in this product: RELAX NG Object Model/Parser; Use of any of this software is governed by the terms of the license below:

The MIT License

Copyright (c)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights touse, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHERIN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

%% The following software may be included in this product: XFree86-VidMode Extension; Use of any of this software is governed by the terms of the license below:

Version 1.1 of
XFree86® ProjectLicence.

Copyright (C) 1994-2004 The XFree86® Project, Inc. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copyof this software and associated documentation files (the "Software"), to deal inthe Software without restriction, including without limitation the rights touse, copy, modify, merge, publish, distribute, sublicense, and/or sell copies ofthe Software, and to permit persons to whom the Software is furnished to do so,subject to the following conditions:

1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution, and in the same place and form as other copyright, license and disclaimer information.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by The XFree86 Project, Inc (http://www.xfree86.org/) and its contributors", in the same place and form as other third-party acknowledgments. Alternately, this acknowledgment may appear in the software itself, in the same form and location as other such third-party acknowledgments.

4. Except as contained in this notice, the name of The XFree86 Project, Inc shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from TheXFree86 Project, Inc.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES

OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE XFREE86 PROJECT, INC OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

%% The following software may be included in this product: RelaxNGCC; Use of any of this software is governed by the terms of the license below:

This is version 2003-May-08 of the Info-ZIP copyright and license. The definitive version of this document should be available at ftp://ftp.info-zip.org/pub/infozip/license.html indefinitely.

Copyright (c) 1990-2003 Info-ZIP. All rights reserved.

For the purposes of this copyright and license, "Info-ZIP" is defined as the following set of individuals:

Mark Adler, John Bush, Karl Davis, Harald Denker, Jean-Michel Dubois, Jean-loup Gailly, Hunter Goatley, Ian Gorman, Chris Herborth, Dirk Haase, Greg Hartwig, Robert Heath, Jonathan Hudson, Paul Kienitz, David Kirschbaum, Johnny Lee, Onno van der Linden, Igor Mandrichenko, Steve P. Miller, Sergio Monesi, Keith Owens, George Petrov, Greg Roelofs, Kai Uwe Rommel, Steve Salisbury, Dave Smith, Christian Spieler, Antoine Verheijen, Paul von Behren, Rich Wales, Mike White

This software is provided "as is," without warranty of any kind, express or implied. In no event shall Info-ZIP or its contributors be held liable for any direct, indirect, incidental, special or consequential damages arising out of the use of or inability to use this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. Redistributions of source code must retain the above copyright notice, definition, disclaimer, and this list of conditions.

2. Redistributions in binary form (compiled executables) must reproduce the above copyright notice, definition, disclaimer, and this list of conditions in documentation and/or other materials provided with the distribution. The sole exception to this condition is redistribution of a standard UnZipSFX binary (including SFXWiz) as part of a self-extracting archive; that is permitted without inclusion of this license, as long as the normal SFX banner has not been removed from the binary or disabled.

3. Altered versions--including, but not limited to, ports to new operating systems, existing ports with new graphical interfaces, and dynamic, shared, or static library versions--must be plainly marked as such and must not be misrepresented as being the original source. Such altered versions also must not be misrepresented as being Info-ZIP releases--including, but not limited to, labeling of the altered versions with the names "Info-ZIP" (or any variation thereof, including, but not limited to, different capitalizations), "Pocket UnZip," "WiZ" or "MacZip" without the explicit permission of Info-ZIP. Such altered versions are further prohibited from misrepresentative use of the Zip-Bugs or Info-ZIP e-mail addresses or of the Info-ZIP URL(s).

4. Info-ZIP retains the right to use the names "Info-ZIP," "Zip," "UnZip," "UnZipSFX," "WiZ," "Pocket UnZip," "Pocket Zip," and

"MacZip" for its own source and binary releases.

%% The following software may be included in this product: XML Security; Use of any of this software is governed by the terms of the license below:

%% The following software may be included in this product: Regexp, Regular Expression Package v. 1.2; Use of any of this software is governed by the terms of the license below:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (http://www.apache.org/)." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. The names "Apache" and "Apache Software Foundation" and "Apache Turbine" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org.

5. Products derived from this software may not be called "Apache", "Apache Turbine", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

==

This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation. For more information on the Apache Software Foundation, please see

http://www.apache.org.

%% The following software may be included in this product: Visual Studio. Use of any of this software is governed by the terms of the license below:

END-USER LICENSE AGREEMENT FOR MICROSOFT SOFTWARE

IMPORTANT-READ CAREFULLY: This End-User License Agreement ("EULA") is a legal agreement between you (either an individual or a single entity) and Microsoft Corporation ("Microsoft) for the Microsoft software that accompanies this EULA, which includes computer software and may include associated media, printed materials, "online" or electronic documentation, and Internet-based services ("Software"). An amendment or addendum to this EULA may accompany the Software. YOU AGREE TO BE BOUND BY THE TERMS OF THIS EULA BY INSTALLING, COPYING, OR OTHERWISE USING THE SOFTWARE. IF YOU DO NOT AGREE, DO NOT INSTALL, COPY, OR USE THE SOFTWARE; YOU MAY RETURN IT TO YOUR PLACE OF PURCHASE (IF APPLICABLE) FOR A FULL REFUND.

MICROSOFT SOFTWARE LICENSE

1. GRANTS OF LICENSE. Microsoft grants you the rights described in this EULA provided that you comply with all terms and conditions of this

EULA. NOTE: Microsoft is not licensing to you any rights with respect to Crystal Reports for Microsoft Visual Studio .NET; your use of Crystal Reports for Microsoft Visual Studio .NET is subject to your acceptance of the terms and conditions of the enclosed (hard copy) end user license agreement from Crystal Decisions for that product.

1.1 General License Grant. Microsoft grants to you as an individual, a personal, nonexclusive license to use the Software, and to make and use copies of the Software for the purposes of designing, developing, testing, and demonstrating your software product(s), provided that you are the only individual using the Software. If you are an entity, Microsoft grants to you a personal, nonexclusive license to use the Software, and to make and use copies of the Software, provided that for each individual using the Software within your organization, you have acquired a separate and valid license for each such individual.

1.2 Documentation. You may make and use an unlimited number of copies of any documentation, provided that such copies shall be used only for personal purposes and are not to be republished or distributed (either in hard copy or electronic form) beyond your premises.

1.3 Storage/Network Use. You may also store or install a copy of the Software on a storage device, such as a network server, used only to install or run the Software on computers used by licensed end users in accordance with Section 1.1. A single license for the Software may not be shared or used concurrently by multiple end users.

1.4 Visual Studio: Effect of EULA. As a suite of development tools and other Microsoft software programs (each such tool or software program, a "Component"), Components that you receive as part of the Software may include a separate end-user license agreement (each, a "Component EULA"). Except as provided in Section 4 ("Prerelease Code"), in the event of inconsistencies between this EULA and any Component EULA, the terms of this EULA shall control. The Software may also contain third-party software programs. Any such software is provided for your use as a convenience and your use is subject to the terms and conditions of any license agreement contained in that software.

2. ADDITIONAL LICENSE RIGHTS -- REDISTRIBUTABLE CODE. In addition to the rights granted in Section 1, certain portions of the Software, as described in this Section 2, are provided to you with additional license rights. These additional license rights are conditioned Everett VSPro 1 Final 11.04.02 upon your compliance with the distribution requirements and license limitations described in Section 3.

2.1 Sample Code. Microsoft grants you a limited, nonexclusive, royalty-free license to: (a) use and modify the source code version of those portions of the Software identified as "Samples" in REDIST.TXT or elsewhere in the Software ("Sample Code") for the sole purposes of designing, developing, and testing your software product(s), and (b) reproduce and distribute the Sample Code, along with any modifications thereof, in object and/or source code form. For applicable redistribution requirements for Sample Code, see Section 3.1 below.

2.2 Redistributable Code: General. Microsoft grants you a limited, nonexclusive, royalty-free license to reproduce and distribute the object code form of any portion of the Software listed in REDIST.TXT ("Redistributable Code"). For general redistribution requirements for Redistributable Code, see Section 3.1 below.

2.3 Redistributable Code: Microsoft Merge Modules ("MSM"). Microsoft grants you a limited, nonexclusive, royalty-free license to reproduce and distribute the content of MSM file(s) listed in REDIST.TXT in the manner described in the Software documentation only so long as you redistribute such content in its entirety and do not modify such content in any way. For all other applicable redistribution requirements for MSM files, see Section 3.1 below.

2.4 Redistributable Code: Microsoft Foundation Classes (MFC), Active Template Libraries (ATL), and C runtimes (CRTs). In addition to the rights granted in Section 1, Microsoft grants you a license to use and modify the source code version of those portions of the Software that are identified as MFC, ATL, or CRTs (collectively, the "VC Redistributables"), for the sole purposes of designing, developing, and testing your software product(s). Provided you comply with Section 3.1 and you rename any files created by you that are included in the Licensee Software (defined below), Microsoft grants you a limited, nonexclusive, royalty-free license to reproduce and distribute the object code version of the VC Redistributables, including any modifications you make. For purposes of this section, "modifications" shall mean enhancements to the functionality of the VC Redistributables. For all other applicable redistribution requirements for VC Redistributables, see Section 3.1 below.

3. DISTRIBUTION REQUIREMENTS AND OTHER LICENSE RIGHTS AND LIMITATIONS. If you choose to exercise your rights under Section 2, any redistribution by you is subject to your compliance with Section 3.1; some of the Redistributable Code has additional limited use rights described in Section 3.2.

3.1 General Distribution Requirements.
(a) If you choose to redistribute Sample Code, or Redistributable Code (collectively, the "Redistributables") as described in Section 2, you agree: (i) except as otherwise noted in Section 2.1 (Sample Code), to distribute the Redistributables only in object code form and in conjunction with and as a part of a software application product developed by you that adds significant and primary functionality to the Redistributables ("Licensee Software"); (ii) that the Redistributables only operate in conjunction with Microsoft Windows platforms; (iii) that if the Licensee Software is distributed beyond Licensee's premises or externally from Licensee's organization, to distribute the Licensee Software containing the Redistributables pursuant to an end user license agreement (which may be "break-the-seal", "click-wrap" or signed), with terms no less protective than those contained in this EULA; (iv) not to use Microsoft's name, logo, or trademarks to market the Licensee Software; (v) to display your own valid copyright notice which shall be sufficient to protect Microsoft's copyright in the Software; Everett VSPro 2 Final 11.04.02 (vi) not to remove or obscure any copyright, trademark or patent notices that appear on the Software as delivered to you; (vii) to indemnify, hold harmless, and defend Microsoft from and against any claims or lawsuits, including attorney's fees, that arise or result from the use or distribution of the Licensee Software; (viii) to otherwise comply with the terms of this EULA; and (ix) agree that Microsoft reserves all rights not expressly granted. You also agree not to permit further distribution of the Redistributables by your end users except you may permit further redistribution of the Redistributables by your distributors to your end-user customers if your distributors only distribute the Redistributables in conjunction with, and as part of, the Licensee Software, you comply with all other terms of this EULA, and your distributors comply with all restrictions of this EULA that are applicable to you.

(b) If you use the Redistributables, then in addition to your compliance with the applicable distribution requirements described for the Redistributables, the following also applies. Your license rights to the Redistributables are conditioned upon your not (i) creating derivative works of the Redistributables in any manner that would cause the Redistributables in whole or in part to become subject to any of the terms of an Excluded License; or (ii) distributing the Redistributables (or derivative works thereof) in any manner that would cause the Redistributables to become subject to any of the terms of an Excluded License. An "Excluded License" is any license that requires as a condition of use, modification and/or distribution of software subject to the Excluded License, that such software or other software combined and/or distributed with such software be (x) disclosed or distributed in source code form; (y) licensed for the purpose of making derivative works; or (z) redistributable at no charge.

3.2 Additional Distribution Requirements for Certain Redistributable Code. If you choose to redistribute the files discussed in this Section, then in addition to the terms of Section 3.1, you must ALSO comply with the following.

(a) Microsoft SQL Server Desktop Engine ("MSDE"). If you redistribute MSDE you agree to comply with the following additional requirements: (a) Licensee Software shall not substantially duplicate the capabilities of Microsoft Access or, in the reasonable opinion of Microsoft, compete with same; and (b) unless Licensee Software requires your customers to license Microsoft Access in order to operate, you shall not reproduce or use MSDE for commercial distribution in conjunction with a general purpose word processing, spreadsheet or database management software product, or an integrated work or product suite whose components include a general purpose word processing, spreadsheet, or database management software product except for the exclusive use of importing data to the various formats supported by Microsoft Access. A product that includes limited word processing, spreadsheet or database components along with other components which provide significant and primary value, such as an accounting product with limited spreadsheet capability, is not considered to be a "general purpose" product.

(b) Microsoft Data Access Components. If you redistribute the Microsoft Data Access Component file identified as MDAC_TYP.EXE, you also agree to redistribute such file in object code only in conjunction with and as a part of a Licensee Software developed by you with a Microsoft development tool product that adds significant and primary functionality to MDAC_TYP.EXE. Everett VSPro 3 Final 11.04.02

3.3 Separation of Components. The Software is licensed as a single product. Its component parts may not be separated for use by more than one user.

3.4 Benchmark Testing. The Software may contain the Microsoft .NET Framework. You may not disclose the results of any benchmark test of the .NET Framework component of the Software to any third party without Microsoft's prior written approval.

4. PRERELEASE CODE. Portions of the Software may be identified as prerelease code ("Prerelease Code"). Such Prerelease Code is not at the level of performance and compatibility of the final, generally available product offering. The Prerelease Code may not operate correctly and may be substantially modified prior to first commercial shipment. Microsoft is not obligated to make this or any later version of the Prerelease Code commercially available. The grant of license to use Prerelease Code expires upon availability of a commercial release of the Prerelease Code from Microsoft. NOTE: In the event that Prerelease Code contains a separate end-user license agreement, the terms and conditions of such end-user license agreement shall govern your use of the corresponding Prerelease Code.

5. RESERVATION OF RIGHTS AND OWNERSHIP. Microsoft reserves all rights not expressly granted to you in this EULA. The Software is protected by copyright and other intellectual property laws and treaties. Microsoft or its suppliers own the title, copyright, and other intellectual property rights in the Software. The Software is licensed, not sold.

6. LIMITATIONS ON REVERSE ENGINEERING, DECOMPILATION, AND DISASSEMBLY. You may not reverse engineer, decompile, or disassemble the Software, except and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation.

7. NO RENTAL/COMMERCIAL HOSTING. You may not rent, lease, lend or provide commercial hosting services with the Software.

8. CONSENT TO USE OF DATA. You agree that Microsoft and its affiliates may collect and use technical information gathered as part of the

product support services provided to you, if any, related to the Software. Microsoft may use this information solely to improve our products or to provide customized services or technologies to you and will not disclose this information in a form that personally identifies you.

9. LINKS TO THIRD PARTY SITES. You may link to third party sites through the use of the Software. The third party sites are not under the control of Microsoft, and Microsoft is not responsible for the contents of any third party sites, any links contained in third party sites, or any changes or updates to third party sites. Microsoft is not responsible for webcasting or any other form of transmission received from any third party sites. Microsoft is providing these links to third party sites to you only as a convenience, and the inclusion of any link does not imply an endorsement by Microsoft of the third party site.

10. ADDITIONAL SOFTWARE/SERVICES. This EULA applies to updates, supplements, add-on components, or Internet-based services components, of the Software that Microsoft may provide to you or make available to you after the date you obtain your initial copy of the Software, unless we provide other terms along with the update, supplement, add-on component, or Internet-based services component. Microsoft reserves the right to discontinue any Internet-based services provided to you or made available to you through the use of the Software.

11. UPGRADES/DOWNGRADES
Everett VSPro 4
Final 11.04.02

11.1 Upgrades. To use a version of the Software identified as an upgrade, you must first be licensed for the software identified by Microsoft as eligible for the upgrade. After upgrading, you may no longer use the software that formed the basis for your upgrade eligibility.

11.2 Downgrades. Instead of installing and using the Software, you may install and use copies of an earlier version of the Software, provided that you completely remove such earlier version and install the current version of the Software within a reasonable time. Your use of such earlier version shall be governed by this EULA, and your rights to use such earlier version shall terminate when you install the Software.

11.3 Special Terms for Version 2003 Upgrade Editions of the Software. If the Software accompanying this EULA is the version 2003 edition of the Software and you have acquired it as an upgrade from the corresponding "2002" edition of the Microsoft software product with the same product name as the Software (the "Qualifying Software"), then Section 11.1 does not apply to you. Instead, you may continue to use the Qualifying Software AND the version 2003 upgrade for so long as you continue to comply with the terms of this EULA and the EULA governing your use of the Qualifying Software. Qualifying Software does not include non-Microsoft software products.

12. NOT FOR RESALE SOFTWARE. Software identified as "Not For Resale" or "NFR," may not be sold or otherwise transfered for value, or used for any purpose other than demonstration, test or evaluation.

13. ACADEMIC EDITION SOFTWARE. To use Software identified as "Academic Edition" or "AE," you must be a "Qualified Educational User." For qualification-related questions, please contact the Microsoft Sales Information Center/One Microsoft Way/Redmond, WA 98052-6399 or the Microsoft subsidiary serving your country.

14. EXPORT RESTRICTIONS. You acknowledge that the Software is subject to U.S. export jurisdiction. You agree to comply with all applicable international and national laws that apply to the Software, including the U.S. Export Administration Regulations, as well as end-user, end- use, and destination restrictions issued by U.S. and other governments. For additional information see <http://www.microsoft.com/exporting/>.

15. SOFTWARE TRANSFER. The initial user of the Software may make a one-time permanent transfer of this EULA and Software to another end user, provided the initial user retains no copies of the Software. This transfer must include all of the Software (including all component parts, the media and printed materials, any upgrades (including any Qualifying Software as defined in Section 11.3), this EULA, and, if applicable, the Certificate of Authenticity). The transfer may not be an indirect transfer, such as a consignment. Prior to the transfer, the end user receiving the Software must agree to all the EULA terms.

16. TERMINATION. Without prejudice to any other rights, Microsoft may terminate this EULA if you fail to comply with the terms and conditions of this EULA. In such event, you must destroy all copies of the Software and all of its component parts. Everett VSPro 5 Final 11.04.02

17. LIMITED WARRANTY FOR SOFTWARE ACQUIRED IN THE US AND CANADA. Except for the "Redistributables," which are provided AS IS without warranty of any kind, Microsoft warrants that the Software will perform substantially in accordance with the accompanying materials for a period of ninety (90) days from the date of receipt.

If an implied warranty or condition is created by your state/jurisdiction and federal or state/provincial law prohibits disclaimer of it, you also have an implied warranty or condition, BUT ONLY AS TO DEFECTS DISCOVERED DURING THE PERIOD OF THIS LIMITED WARRANTY (NINETY DAYS). AS TO ANY DEFECTS DISCOVERED AFTER THE NINETY-DAY PERIOD, THERE IS NO WARRANTY OR CONDITION OF ANY KIND.

Some states/jurisdictions do not allow limitations on how long an implied warranty or condition lasts, so the above limitation may not apply to you. Any supplements or updates to the Software, including without limitation, any (if any) service packs or hot fixes provided to you after the expiration of the ninety day Limited Warranty period are not covered by any warranty or condition, express, implied or statutory.

LIMITATION ON REMEDIES; NO CONSEQUENTIAL OR OTHER DAMAGES. Your exclusive remedy for any breach of this Limited Warranty is as set forth below. Except for any refund elected by Microsoft, YOU ARE NOT ENTITLED TO ANY DAMAGES, INCLUDING BUT NOT LIMITED TO CONSEQUENTIAL DAMAGES, if the Software does not meet Microsoft's Limited Warranty, and, to the maximum extent allowed by applicable law, even if any remedy fails of its essential purpose. The terms of Section 19 ("Exclusion of Incidental, Consequential and Certain Other Damages") are also incorporated into this Limited Warranty. Some states/jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you. This Limited Warranty gives you specific legal rights. You may have other rights which vary from state/jurisdiction to state/jurisdiction. YOUR EXCLUSIVE REMEDY. Microsoft's and its suppliers' entire liability and your exclusive remedy for any breach of this Limited Warranty or for any other breach of this EULA or for any other liability relating to the Software shall be, at Microsoft's option from time to time exercised subject to applicable law, (a) return of the amount paid (if any) for the Software, or (b) repair or replacement of the Software, that does not meet this Limited Warranty and that is returned to Microsoft with a copy of your receipt. You will receive the remedy elected by Microsoft without charge, except that you are responsible for any expenses you may incur (e.g. cost of shipping the Software to Microsoft). This Limited Warranty is void if failure of the Software has resulted from accident, abuse, misapplication, abnormal use or a virus. Any replacement Software will be warranted for the remainder of the original warranty period or thirty (30) days, whichever is longer, and Microsoft will use commercially reasonable efforts to provide your remedy within a commercially reasonable time of your compliance with Microsoft's warranty remedy procedures. Outside the United States or Canada, neither these remedies nor any product support services offered by Microsoft are available without proof of purchase from an authorized international source. To exercise your remedy, contact: Microsoft, Attn. Microsoft Sales Information Center/One Microsoft Way/Redmond, WA 98052-6399, or the Microsoft subsidiary serving your country.

18. DISCLAIMER OF WARRANTIES. The Limited Warranty that appears above is the only express warranty made to you and is provided in lieu of any other express warranties or similar obligations (if any) created by any advertising, documentation, packaging, or other communications. EXCEPT FOR THE LIMITED WARRANTY AND TO THE MAXIMUM Everett VSPro 6 Final 11.04.02 EXTENT PERMITTED BY APPLICABLE LAW, MICROSOFT AND ITS SUPPLIERS PROVIDE THE SOFTWARE AND SUPPORT SERVICES (IF ANY) AS IS AND WITH ALL FAULTS, AND HEREBY DISCLAIM ALL OTHER WARRANTIES AND CONDITIONS, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, ANY (IF ANY) IMPLIED WARRANTIES, DUTIES OR CONDITIONS OF MERCHANTABILITY, OF FITNESS FOR A PARTICULAR PURPOSE, OF RELIABILITY OR AVAILABILITY, OF ACCURACY OR COMPLETENESS OF RESPONSES, OF RESULTS, OF WORKMANLIKE EFFORT, OF LACK OF VIRUSES, AND OF LACK OF NEGLIGENCE, ALL WITH REGARD TO THE SOFTWARE, AND THE PROVISION OF OR FAILURE TO PROVIDE SUPPORT OR OTHER SERVICES, INFORMATION, SOFTWARE, AND RELATED CONTENT THROUGH THE SOFTWARE OR OTHERWISE ARISING OUT OF THE USE OF THE SOFTWARE. ALSO, THERE IS NO WARRANTY OR CONDITION OF TITLE, QUIET ENJOYMENT, QUIET POSSESSION, CORRESPONDENCE TO DESCRIPTION OR NON-INFRINGEMENT WITH REGARD TO THE SOFTWARE.

19. EXCLUSION OF INCIDENTAL, CONSEQUENTIAL AND CERTAIN OTHER DAMAGES. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL MICROSOFT OR ITS SUPPLIERS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, PUNITIVE, INDIRECT, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING, BUT NOT LIMITED TO, DAMAGES FOR LOSS OF PROFITS OR CONFIDENTIAL OR OTHER INFORMATION, FOR BUSINESS INTERRUPTION, FOR PERSONAL INJURY, FOR LOSS OF PRIVACY, FOR FAILURE TO MEET ANY DUTY INCLUDING OF GOOD FAITH OR OF REASONABLE CARE, FOR NEGLIGENCE, AND FOR ANY OTHER PECUNIARY OR OTHER LOSS WHATSOEVER) ARISING OUT OF OR IN ANY WAY RELATED TO THE USE OF OR INABILITY TO USE THE SOFTWARE, THE PROVISION OF OR FAILURE TO PROVIDE SUPPORT OR OTHER SERVICES, INFORMATION, SOFTWARE, AND RELATED CONTENT THROUGH THE SOFTWARE OR OTHERWISE ARISING OUT OF THE USE OF THE SOFTWARE, OR OTHERWISE UNDER OR IN CONNECTION WITH ANY PROVISION OF THIS EULA, EVEN IN THE EVENT OF THE FAULT, TORT (INCLUDING NEGLIGENCE), MISREPRESENTATION, STRICT LIABILITY, BREACH OF CONTRACT OR BREACH OF WARRANTY OF MICROSOFT OR ANY SUPPLIER, AND EVEN IF MICROSOFT OR ANY SUPPLIER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

20. LIMITATION OF LIABILITY AND REMEDIES. NOTWITHSTANDING ANY DAMAGES THAT YOU MIGHT INCUR FOR ANY REASON WHATSOEVER (INCLUDING, WITHOUT LIMITATION, ALL DAMAGES REFERENCED HEREIN AND ALL DIRECT OR GENERAL DAMAGES IN CONTRACT OR ANYTHING ELSE), THE ENTIRE LIABILITY OF MICROSOFT AND ANY OF ITS SUPPLIERS UNDER ANY PROVISION OF THIS EULA AND YOUR EXCLUSIVE REMEDY HEREUNDER (EXCEPT FOR ANY REMEDY OF REPAIR OR REPLACEMENT ELECTED BY MICROSOFT WITH RESPECT TO ANY BREACH OF THE LIMITED WARRANTY) SHALL BE LIMITED TO THE GREATER OF THE ACTUAL DAMAGES YOU INCUR IN REASONABLE RELIANCE ON THE SOFTWARE UP TO THE AMOUNT ACTUALLY PAID BY YOU FOR THE SOFTWARE OR US$5.00. THE FOREGOING LIMITATIONS, EXCLUSIONS AND DISCLAIMERS (INCLUDING SECTIONS 17, 18, AND 19) SHALL APPLY TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, EVEN IF ANY REMEDY FAILS ITS ESSENTIAL PURPOSE. Everett VSPro 7 Final 11.04.02

21. U.S. GOVERNMENT LICENSE RIGHTS. All Software provided to the U.S. Government pursuant to solicitations issued on or after December 1, 1995 is provided with the commercial license rights and restrictions described elsewhere herein. All Software provided to the U.S. Government pursuant to solicitations issued prior to December 1, 1995 is provided with "Restricted Rights" as provided for in FAR, 48 CFR 52.227-14 (JUNE 1987) or DFAR, 48 CFR 252.227-7013 (OCT 1988), as applicable.

22. APPLICABLE LAW. If you acquired this Software in the United States, this EULA is governed by the laws of the State of Washington. If you acquired this Software in Canada, unless expressly prohibited by local law, this EULA is governed by the laws in force in the Province of Ontario, Canada; and, in respect of any dispute which may arise hereunder, you consent to the jurisdiction of the federal and provincial courts

sitting in Toronto, Ontario. If you acquired this Software in the European Union, Iceland, Norway, or Switzerland, then local law applies. If you acquired this Software in any other country, then local law may apply.

23. ENTIRE AGREEMENT; SEVERABILITY. This EULA (including any addendum or amendment to this EULA which is included with the Software) are the entire agreement between you and Microsoft relating to the Software and the support services (if any) and they supersede all prior or contemporaneous oral or written communications, proposals and representations with respect to the Software or any other subject matter covered by this EULA. To the extent the terms of any Microsoft policies or programs for support services conflict with the terms of this EULA, the terms of this EULA shall control. If any provision of this EULA is held to be void, invalid, unenforceable or illegal, the other provisions shall continue in full force and effect.

Si vous avez acquis votre produit Microsoft au CANADA, la garantie limitée suivante s'applique :

GARANTIE LIMITÉE

Sauf pur celles du "Redistributables," qui sont fournies "comme telles," Microsoft garantit que le Logiciel fonctionnera conformément aux documents inclus pendant une période de 90 jours suivant la date de réception.

Si une garantie ou condition implicite est créée par votre État ou votre territoire et qu'une loifédérale ou provinciale ou d'un État en interdit le déni, vous jouissez également d'une garantie ou condition implicite, MAIS UNIQUEMENT POUR LES DÉFAUTS DÉCOUVERTS DURANT LA PÉRIODE DE LA PRÉSENTE GARANTIE LIMITÉE (QUATRE-VINGT-DIX JOURS). IL N'Y A AUCUNE GARANTIE OU CONDITION DE QUELQUE NATURE QUECE SOIT QUANT AUX DÉFAUTS DÉCOUVERTS APRÈS CETTE PÉRIODE DE QUATRE- VINGT-DIX JOURS. Certains États ou territoires ne permettent pas de limiter la durée d'une garantie ou condition implicite de sorte que la limitation ci-dessus peut ne pas s'appliquer à vous.

Tous les suppléments ou toutes les mises à jour relatifs au Logiciel, notamment, les ensembles de services ou les réparations à chaud (le cas échéant) qui vous sont fournis après l'expiration de la période de quatre-vingt-dix jours de la garantie limitée ne sont pas couverts par quelque garantie ou condition que ce soit, expresse, implicite ou en vertu de la loi.

LIMITATION DES RECOURS; ABSENCE DE DOMMAGES INDIRECTS OU AUTRES.

Votre recours exclusif pour toute violation de la présente garantie limitée est décrit ci-après.

Sauf pour tout remboursement au choix de Microsoft, si le Logiciel ne respecte pas la Everett VSPro 8 Final 11.04.02 garantie limitée de Microsoft et, dans la mesure maximale permise par les lois applicables, même si tout recours n'atteint pas son but essentiel, VOUS N'AVEZ DROIT À AUCUNS DOMMAGES, NOTAMMENT DES DOMMAGES INDIRECTS. Les termes de la clause «Exclusion des dommages accessoires, indirects et de certains autres dommages » sontégalement intégrées à la présente garantie limitée. Certains États ou territoires ne permettent pas l'exclusion ou la limitation des dommages indirects ou accessoires de sorte que la limitation ou l'exclusion ci-dessus peut ne pas s'appliquer à vous. La présente garantie limitée vous donne des droits légaux spécifiques. Vous pouvez avoir d'autres droits qui peuvent varier d'unterritoire ou d'un État à un autre. VOTRE RECOURS EXCLUSIF. La seule responsabilité obligation de Microsoft et de ses fournisseurs et votre recours exclusif pour toute violation de la présente garantie limitée ou pour toute autre violation du présent contrat ou pour toute autre responsabilité relative au Logiciel seront, selon le choix de Microsoft exercé de temps à autre sous réserve de toute loi applicable, a) le remboursement du prix payé, le cas échéant, pour le Logiciel ou b) la réparation ou le remplacement du Logiciel qui ne respecte pas la présente garantie limitée et qui est retourné à Microsoft avec une copie de votre reçu. Vous recevrez la compensation choisie par Microsoft, sans frais, sauf que vous êtes responsable des dépenses que vous pourriez engager (p. ex., les frais d'envoi du Logiciel à Microsoft). La présente garantie limitée est nulle si la défectuosité du Logiciel est causée par un accident, un usage abusif, une mauvaise application, un usage anormal ou un virus. Tout Logiciel de remplacement sera garanti pour le reste de la période initiale de la garantie ou pendant trente (30) jours, selon la plus longue entre ces deux périodes. À l'extérieur des États-Unis ou du Canada, ces recours ou l'un quelconque des services de soutien technique offerts par Microsoft ne sont pas disponibles sans preuve d'achat d'une source internationale autorisée. Pour exercer votre recours, vous devez communiquer avec Microsoft et vous adresser au Microsoft Sales Information Center/One Microsoft Way/Redmond, WA 98052-6399, ou à la filiale de Microsoft de votre pays.

DÉNI DE GARANTIES. La garantie limitée qui apparaît ci-dessus constitue la seule garantie expresse qui vous est donnée et remplace toutes autres garanties expresses (s'il en est) crées par une publicité, un document, un emballage ou une autre communication. SAUF EN CE QUI A TRAIT À LA GARANTIE LIMITÉE ET DANS LA MESURE MAXIMALE PERMISE PAR LES LOIS APPLICABLES, LE LOGICIEL ET LES SERVICES DE SOUTIEN TECHNIQUE (LE CAS ÉCHÉANT) SONT FOURNIS TELS QUELS ET AVEC TOUS LES DÉFAUTS PAR MICROSOFT ET SES FOURNISSEURS, LESQUELS PAR LES PRÉSENTES DÉNIENT TOUTES AUTRES GARANTIES ET CONDITIONS EXPRESSES, IMPLICITES OU EN VERTU DE LA LOI, NOTAMMENT, MAIS SANS LIMITATION, (LE CAS ÉCHÉANT) LESGARANTIES, DEVOIRS OU CONDITIONS IMPLICITES DE QUALITÉ MARCHANDE, D'ADAPTATION À UNE FIN PARTICULIÈRE, DE FIABILITÉ OU DE DISPONIBILITÉ, D'EXACTITUDE OU D'EXHAUSTIVITÉ DES RÉPONSES, DES RÉSULTATS, DES EFFORTS DÉPLOYÉS SELON LES RÈGLES DE L'ART, D'ABSENCE DE VIRUS ET D'ABSENCE DE NÉGLIGENCE, LE TOUT À L'ÉGARD DU LOGICIEL ET DE LA PRESTATION OU DE L'OMISSION DE LA PRESTATION DES SERVICES DE SOUTIEN TECHNIQUE OU À L'ÉGARD DE LA FOURNITURE OU DE L'OMISSION DE LA FOURNITURE DE TOUS AUTRES SERVICES, RENSEIGNEMENTS, LOGICIELS, ET CONTENU QUI S'Y RAPPORTE GRÂCE AU LOGICIEL OU PROVENANT AUTREMENT DE L'UTILISATION DU LOGICIEL . PAR AILLEURS, IL N'Y A AUCUNE GARANTIE OU CONDITION QUANT AU TITRE DE PROPRIÉTÉ, À LA JOUISSANCE OU LA POSSESSION PAISIBLE, À LA CONCORDANCE À UNE DESCRIPTION NI QUANT À UNE ABSENCE DE CONTREFAÇON CONCERNANT LE LOGICIEL.

EXCLUSION DES DOMMAGES ACCESSOIRES, INDIRECTS ET DE CERTAINS AUTRES DOMMAGES. DANS LA MESURE MAXIMALE PERMISE PAR LES LOIS APPLICABLES, EN AUCUN CAS MICROSOFT OU SES FOURNISSEURS NE SERONT RESPONSABLES DES DOMMAGES SPÉCIAUX, CONSÉCUTIFS, ACCESSOIRES OU INDIRECTS DE Everett VSPro 9 Final 11.04.02 QUELQUE NATURE QUE CE SOIT (NOTAMMENT, LES DOMMAGES À L'ÉGARD DUMANQUE À GAGNER OU DE LA DIVULGATION DE RENSEIGNEMENTS CONFIDENTIELS OU AUTRES, DE LA PERTE D'EXPLOITATION, DE BLESSURES CORPORELLES, DE LA VIOLATION DE LA VIE PRIVÉE, DE L'OMISSION DE REMPLIR TOUT DEVOIR, Y COMPRIS D'AGIR DE BONNE FOI OU D'EXERCER UN SOIN RAISONNABLE, DE LA NÉGLIGENCE ET DE TOUTE AUTRE PERTE PÉCUNIAIRE OU AUTRE PERTE DE QUELQUE NATURE QUE CE SOIT) SE RAPPORTANT DE QUELQUEMANIÈRE QUE CE SOIT À L'UTILISATION DU LOGICIEL OU À L'INCAPACITÉ DE S'EN SERVIR, À LA PRESTATION OU À L'OMISSION DE LA PRESTATION DE SERVICES DE SOUTIEN TECHNIQUE OU À LA FOURNITURE OU À L'OMISSION DE LA FOURNITURE DE TOUS AUTRES SERVICES, RENSEIGNEMENTS, LOGICIELS, ET CONTENU QUI S'Y RAPPORTE GRÂCE AU LOGICIEL OU PROVENANT AUTREMENT DE L'UTILISATION DU LOGICIEL OU AUTREMENT AUX TERMES DE TOUTE DISPOSITION DE LA PRÉSENTE CONVENTION OU RELATIVEMENT À UNE TELLE DISPOSITION, MÊME EN CAS DE FAUTE, DE DÉLIT CIVIL (Y COMPRIS LANÉGLIGENCE), DE RESPONSABILITÉ STRICTE, DE VIOLATION DE CONTRAT OU DEVIOLATION DE GARANTIE DE MICROSOFT OU DE TOUT FOURNISSEUR ET MÊME SI MICROSOFT OU TOUT FOURNISSEUR A ÉTÉ AVISÉ DE LA POSSIBILITÉ DE TELS DOMMAGES.

LIMITATION DE RESPONSABILITÉ ET RECOURS. MALGRÉ LES DOMMAGES QUE VOUS PUISSIEZ SUBIR POUR QUELQUE MOTIF QUE CE SOIT (NOTAMMENT, MAISSANS LIMITATION, TOUS LES DOMMAGES SUSMENTIONNÉS ET TOUS LES DOMMAGES DIRECTS OU GÉNÉRAUX OU AUTRES), LA SEULE RESPONSABILITÉ DE MICROSOFT ET DE L'UN OU L'AUTRE DE SES FOURNISSEURS AUX TERMES DE TOUTE DISPOSITION DE LA PRÉSENTE CONVENTION ET VOTRE RECOURS EXCLUSIF À L'ÉGARD DE TOUT CE QUI PRÉCÈDE

(SAUF EN CE QUI CONCERNETOUT RECOURS DE RÉPARATION OU DE REMPLACEMENT CHOISI PAR MICROSOFT À L'ÉGARD DE TOUT MANQUEMENT À LA GARANTIE LIMITÉE) SELIMITE AU PLUS ÉLEVÉ ENTRE LES MONTANTS SUIVANTS : LE MONTANT QUE VOUS AVEZ RÉELLEMENT PAYÉ POUR LE LOGICIEL OU 5,00 $US. LES LIMITES, EXCLUSIONS ET DÉNIS QUI PRÉCÈDENT (Y COMPRIS LES CLAUSES CI-DESSUS), S'APPLIQUENT DANS LA MESURE MAXIMALE PERMISE PAR LES LOIS APPLICABLES, MÊME SI TOUT RECOURS N'ATTEINT PAS SON BUT ESSENTIEL.

À moins que cela ne soit prohibé par le droit local applicable, la présente Convention est régie par les lois de la province d'Ontario, Canada. Vous consentez à la compétence des tribunaux fédéraux et provinciaux siégeant à Toronto, dans la province d'Ontario.

Au cas où vous auriez des questions concernant cette license ou que vous désiriez vous mettre en rapport avec Microsoft pour quelque raison que ce soit, veuillez utiliser l'information contenue dans le Logiciel pour contacter la filiale de Microsoft desservant votre pays, ou visitez Microsoft sur le World Wide Web à http://www.microsoft.com.

The following MICROSOFT GUARANTEE applies to you if you acquired this Software in any other country:

Statutory rights not affected -The following guarantee is not restricted to any territory and does not affect any statutory rights that you may have from your reseller or from Microsoft if you acquired the Software directly from Microsoft. If you acquired the Software or any support services in Australia, New Zealand or Malaysia, please see the "Consumer rights" section below.

Everett VSPro 10
Final 11.04.02

The guarantee -The Software is designed and offered as a general-purpose software, not for any user's particular purpose. You accept that no Software is error free and you are strongly advised to back-up your files regularly. Provided that you have a valid license, Microsoft guarantees that a) for a period of 90 days from the date of receipt of your license to use the Software or the shortest period permitted by applicable law it will perform substantially in accordance with the written materials that accompany the Software; and b) any support services provided by Microsoft shall be substantially as described in applicable written materials provided to you by Microsoft and Microsoft support engineers will use reasonable efforts, care and skill to solve any problem issues. In the event that the Software fails to comply with this guarantee, Microsoft will either (a) repair or replace the Software or (b) return the price you paid. This guarantee is void if failure of the Software results from accident, abuse or misapplication. Any replacement Software will be guaranteed for the remainder of the original guarantee period or 30 days, whichever period is longer. You agree that the above guarantee is your sole guarantee in relation to the Software and any support services.

Exclusion of All Other Terms -To the maximum extent permitted by applicable law and subject to the guarantee above, Microsoft disclaims all warranties, conditions and other terms, either express or implied (whether by statute, common law, collaterally or otherwise) including but not limited to implied warranties of satisfactory quality and fitness for particular purpose with respect to the Software and the written materials that accompany the Software. Any implied warranties that cannot be excluded are limited to 90 days or to the shortest period permitted by applicable law, whichever is greater.

Limitation of Liability -To the maximum extent permitted by applicable law and except as provided in the Microsoft Guarantee, Microsoft and its suppliers shall not be liable for any damages whatsoever (including without limitation, damages for loss of business profits, business interruption, loss of business information or other pecuniary loss) arising out of the use or inability to use the Software, even if Microsoft has been advised of the possibility of such damages. In any case

Microsoft's entire liability under any provision of this Agreement shall be limited to the amount actually paid by you for the Software. These limitations do not apply to any liabilities that cannot be excluded or limited by applicable laws.

Consumer rights -Consumers in Australia, New Zealand or Malaysia may have the benefit of certain rights and remedies by reason of the Trade Practices Act and similar state and territory laws in Australia, the Consumer Guarantees Act in New Zealand and the Consumer Protection Act in Malaysia in respect of which liability cannot lawfully be modified or excluded. If you acquired the Software in New Zealand for the purposes of a business, you confirm that the Consumer Guarantees Act does not apply. If you acquired the Software in Australia and if Microsoft breaches a condition or warranty implied under any law which cannot lawfully be modified or excluded by this agreement then, to the extent permitted by law, Microsoft's liability is limited, at Microsoft's option, to: (i) in the case of the Software: a) repairing or replacing the Software; or b) the cost of such repair or replacement; and (ii) in the case of support services: a) re-supply of the services; or b) the cost of having the services supplied again.

Everett VSPro 11
Final 11.04.02

Should you have any questions concerning this EULA, or if you desire to contact Microsoft for any reason, please use the address information enclosed in this Software to contact the Microsoft subsidiary serving your country or visit Microsoft on the World Wide Web at http://www.microsoft.com.

Everett VSPro 12
Final 11.04.02

%% The following software may be included in this product: zlib; Use of any of this software is governed by the terms of the license below:

zlib.h -- interface of the 'zlib' general purpose compression library
version 1.1.3, July 9th, 1998

Copyright (C) 1995-1998 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly	Mark Adler
jloup@gzip.org	madler@alumni.caltech.edu

The data format used by the zlib library is described by RFCs (Request for Comments) 1950 to 1952 in the files ftp://ds.internic.net/rfc/rfc1950.txt (zlib format), rfc1951.txt (deflate format) and rfc1952.txt (gzip format

```
 *
 * 5. Products derived from this software may not be called "OpenSSL"
 *    nor may "OpenSSL" appear in their names without prior written
 *    permission of the OpenSSL Project.
 *
 * 6. Redistributions of any form whatsoever must retain the following
 *    acknowledgment:
 *    "This product includes software developed by the OpenSSL Project
 *    for use in the OpenSSL Toolkit (http://www.openssl.org/)"
 *
 * THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
 * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 * PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE OpenSSL PROJECT OR
 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
 * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
 * OF THE POSSIBILITY OF SUCH DAMAGE.
 * ====================================================================
 *
 * This product includes cryptographic software written by Eric Young
 * (eay@cryptsoft.com).  This product includes software written by Tim
 * Hudson (tjh@cryptsoft.com).
 *
 */

 Original SSLeay License
 -----------------------

/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
 * All rights reserved.
 *
 * This package is an SSL implementation written
 * by Eric Young (eay@cryptsoft.com).
 * The implementation was written so as to conform with Netscapes SSL.
 *
 * This library is free for commercial and non-commercial use as long as
 * the following conditions are aheared to.  The following conditions
 * apply to all code found in this distribution, be it the RC4, RSA,
 * lhash, DES, etc., code; not just the SSL code.  The SSL documentation
 * included with this distribution is covered by the same copyright terms
 * except that the holder is Tim Hudson (tjh@cryptsoft.com).
 *
 * Copyright remains Eric Young's, and as such any Copyright notices in
 * the code are not to be removed.
 * If this package is used in a product, Eric Young should be given attribution
 * as the author of the parts of the library used.
 * This can be in the form of a textual message at program startup or
 * in documentation (online or textual) provided with the package.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 *    must display the following acknowledgement:
 *    "This product includes cryptographic software written by
 *     Eric Young (eay@cryptsoft.com)"
 *    The word 'cryptographic' can be left out if the rouines from the library
 *    being used are not cryptographic related :-).
 * 4. If you include any Windows specific code (or a derivative thereof) from
 *    the apps directory (application code) you must include an acknowledgement:
 *    "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
 *
 * THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *
 * The license and distribution terms for any publically available version or
 * derivative of this code cannot be changed.  i.e. this code cannot simply be
 * copied and put under another distribution license
 * [including the GNU Public Licence.]
 */
```