



**SAP Control Center for SAP Mobile
Platform**

SAP Mobile Platform 2.3 SP04

DOCUMENT ID: DC01929-01-0234-02

LAST REVISED: May 2014

Copyright © 2014 by Sybase, Inc. All rights reserved.

This publication pertains to Sybase software and to any subsequent release until otherwise indicated in new editions or technical notes. Information in this document is subject to change without notice. The software described herein is furnished under a license agreement, and it may be used or copied only in accordance with the terms of that agreement.

Upgrades are provided only at regularly scheduled software release dates. No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of Sybase, Inc.

Sybase trademarks can be viewed at the Sybase trademarks page at <http://www.sybase.com/detail?id=1011207>. Sybase and the marks listed are trademarks of Sybase, Inc. ® indicates registration in the United States of America.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world.

Java and all Java-based marks are trademarks or registered trademarks of Oracle and/or its affiliates in the U.S. and other countries.

Unicode and the Unicode Logo are registered trademarks of Unicode, Inc.

All other company and product names mentioned may be trademarks of the respective companies with which they are associated.

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

Contents

Get Started	1
About SAP Control Center for SAP Mobile Platform	1
Documentation Roadmap for SAP Mobile Platform	1
SAP Mobile Platform Administration by Node	2
Cluster Administration	3
Server Administration	5
Application and User Management Overview	5
Domain Management	7
MBO Package Management Overview	8
Hybrid App Package Management Overview	10
Security Administration Overview	11
System Monitoring Overview	11
Starting and Stopping SAP Control Center in SAP Mobile Platform	13
Cleaning Up the Flash Player Cache	14
Copying and Pasting Properties	14
Getting Started with Production Clusters	14
Getting Started After Installing	15
Configuring Memory Usage	17
Configuring the Automatic Logout Timer	20
Manually Opening the SAP Mobile Platform Console	20
Adding or Updating server-name Registration Properties	21
SAP Mobile Server Registration Properties	21
Understanding the SAP Control Center Interface	22
User Interface Overview	22
Perspectives	24
Views	26
Repository	28

SAP Control Center Console	33
SAP Control Center Security	36
Platform Administration Roles and Tasks	37
SAP Mobile Platform Administrator	37
Platform Administration Tasks	37
SAP Mobile Platform Domain Administrator	38
Domain Administration Tasks	39
SAP Mobile Platform Helpdesk	39
Help Desk Operator Tasks	40
Administer	41
Clusters	41
Cluster-Affecting Configuration Changes	41
Cluster Properties	41
Viewing Cluster Information	64
Checking System Licensing Information	64
Checking Cluster Status	65
Managing Service Pack Information	65
Sharing Cluster Information With SAP Servers ..	66
Relay Server	71
Configuring SAP Mobile Server to use Relay	
Server	72
Managing Configured Relay Servers	80
Relay Server Tab Reference	84
SAP Mobile Server	84
Server List	85
Relay Server Outbound Enabler	88
Server Log	96
Domains	100
Creating and Enabling a New Domain	101
Deleting a Domain	102
Registering a Domain Administrator User	103
Assigning Domain Administrators to a Domain .	104
Viewing Applications for a Domain	105
Viewing Application Connections for a Domain .	105

Scheduling Accumulated Data Cleanup for	
Domains	107
Domain Logs	110
Client Logs	162
Connections	167
Configuring Domain Security	195
Security Configurations	199
Creating a Security Configuration	200
Assigning a Security Configuration to a Domain	
.....	246
Viewing Security Configuration Usage	246
Anonymous Access Security Configuration	247
Agency Security Configuration	247
SiteMinder Authentication with SAP Mobile	
Platform	247
Applications	253
Activating and Maintaining Applications	254
Defining Applications	254
Maintaining Activated Applications	302
Transporting Applications Between	
Environments Using Export and Import	315
Application Connections	318
Application Connection Templates	326
Application Connection Properties	329
Deploy	339
MBO Packages	339
Deploying MBO Packages	340
MBO Package Management	348
MBO Subscription Management	357
Reviewing MBO History	363
Reviewing Operation History	363
DOE-C Packages	364
Deploying and Configuring DOE-C Packages	364
Viewing and Changing Package Connection	
Properties	365

Setting the Bulk Load Timeout Property	365
Checking and Resolving DOE-C User Failures .	366
Package Subscription Properties	366
Hybrid App Packages	367
Deploying Hybrid App Packages	367
Enabling and Configuring the Notification	
Mailbox	371
Configuring a Hybrid App Package	372
Agency Applications	380
Creating Agency Application Definitions	381
Deploying Agency Application Files to an	
Existing Application	382
Downloading Agency Application Metadata for	
Deploying	383
Transporting Applications Between	
Environments Using Export and Import	383
Monitor	387
Monitoring Usage	387
System Monitoring Overview	388
Monitoring Configuration	390
Configuring Monitoring Performance Properties	
.....	390
Monitoring Profiles	392
Creating and Enabling a Monitoring Profile	392
Monitoring Data	394
Reviewing System Monitoring Data	394
Purging Monitoring Data	395
Exporting Monitoring Data	395
Searching Monitoring Data	396
Viewing Package-Level Cache Statistics	397
Monitoring Data Categories	397
Index	421

Get Started

Set up SAP® Control Center.

About SAP Control Center for SAP Mobile Platform

SAP Control Center provides a single comprehensive Web administration console to configure and manage SAP products and their components.

SAP Control Center combines a modular architecture, a rich administrative console, agents, common services, and tools for managing and controlling SAP products. SAP Mobile Platform is one of many SAP products that use SAP Control Center as its management and administrative tool.

As part of an SAP Mobile Platform installation, SAP Control Center can be used in three ways:

- In a personal development environment, developers may act as administrators to set up a personal testing environment. Development administrators use SAP Control Center to deploy and configure packages, register messaging devices, and so on. No other additional configuration or administration may be required.
- In a distributed or shared development environment, administrators use SAP Control Center to set up an SAP Mobile Server, manage packages, manage devices, configure Hybrid App packages, as well as review server and domain logs, and monitoring-related data.
- In a production environment, administrators use SAP Control Center on a regular basis to perform the same tasks described for a shared development environment. They also configure the operation of SAP Mobile Servers, and administer day-to-day activities of the production environment. Administrators must also routinely monitor the overall health and performance of the system, which may include clusters and domains.

Documentation Roadmap for SAP Mobile Platform

SAP® Mobile Platform documents are available for administrative and mobile development user roles. Some administrative documents are also used in the development and test environment; some documents are used by all users.

See *Documentation Roadmap* in *Fundamentals* for document descriptions by user role.

Check the Product Documentation Web site regularly for updates: <http://sybooks.sybase.com/sybooks/sybooks.xhtml?id=1289&c=firsttab&a=0&p=categories>, then navigate to the most current version.

SAP Mobile Platform Administration by Node

The left navigation pane in the SAP Control Center console displays a tree of administrable features in the form of nodes, some of which can be expanded to reveal a more granular view of the cluster environment. These nodes let you manage and configure the main components of SAP Mobile Platform.

Clicking nodes allows you to administer the following features through SAP Control Center. However, be aware of the following dependencies:

- There are three administration roles. Users with the platform administration role have access to all nodes. Users with the domain administrator role see only the "Domains" nodes for their assigned domains. Users with the help desk role has read only access to system configuration values.
- You must have the correct SAP Mobile Platform version and license for these nodes to be functional when they are visible.

Node	Purpose
Cluster	View general cluster and server node properties and access the server list for the cluster. Additionally perform configuration for Relay Server, System Landscape Directory (SLD) server, and SAP Licence Audit.
Domains	Add, delete, enable, and disable domains. Expand this node to manage the security, package, role mappings, cache group, synchronization group, subscription, and connection configurations for each domain. You can also expand the Applications subnode to see the applications and application connections managed from the domain.
Servers	Configure Java Virtual Machine properties for each server.
Applications	Add, view, delete, and edit applications, application users, application connections, and application connection template operations as part of application activation.
Security	Add, view, edit, and delete domain administrators. Add or delete a security configuration. Each security configuration contains one or more security providers for authentication, authorization, attribution and auditing. Once configured, security configurations can be assigned to domains and then mapped to one or more packages, depending on the requirements for each.
Hybrid App	Deploy and manage Hybrid App packages and configure the notification mailbox. Deployed Hybrid App packages are listed below this node. Use the individual Hybrid App nodes to manage Hybrid App package properties, matching rules, context variables, error logs, application connections, and, optionally, queue items.

Node	Purpose
Monitoring	Create and manage settings for monitoring security, replication synchronization, messaging synchronization, device notification, data change notification, queue, package, user, and cache activities.

Cluster Administration

The goal of cluster administration is to ensure that clusters and servers work smoothly, and scale over time. By default, the SAP Mobile Platform is installed as a one-node cluster. The one-node cluster is supported in development or test environments. Production deployments of SAP Mobile Platform are likely to require multiple nodes. Cluster administration is mostly a nonroutine administration task.

See *Designing the Landscape* in *Landscape Design and Integration*.

Table 1. Cluster administration tasks

Task	Frequency	Accomplished by
Installing the cluster	One-time installation per cluster	SAP Mobile Platform installer
Setting up Relay Servers	One-time initial installation and configuration; occasionally adding servers to the cluster	Manual installation; manual set-up using configuration files.
Suspending and resuming server nodes	On demand, as required	SAP Control Center
Setting cluster properties, including cache database settings, monitoring database setup, and so on	Once, or as cluster changes require	Manual configuration using files and .BAT scripts.
Tuning the server performance.	Post installation with infrequent tuning as required	SAP Control Center.

Task	Frequency	Accomplished by
Configuring the cluster to: • Set the configuration cache properties • Set the solution manager URL • Set the replication ports and properties • Set the messaging synchronization ports • Set the amount of data that can be downloaded through an HTTP connection using the BlackBerry MDS • Set the management ports for communication requests from SAP Control Center • Configure the client dispatcher • Set how applications handle DCN requests sent through an HTTP GET operation • Create security profiles for secure communication • Set up secure synchronization • Tune server performance	Post installation configuration with infrequent tuning as required	
Setting cluster log file settings for SAP Mobile Server system components	Once, unless log data requirements change	
Administering the runtime databases	Routine to ensure that the database server is monitored and backed up, that there is sufficient space for SAP Mobile Platform metadata and cached data tables, and that performance is within acceptable limits (performance tuning)	Established processes and command line utilities. Consult with your database administrator.

Task	Frequency	Accomplished by
Reviewing licensing information, including currently used licenses count	Occasional, or as device user registration and deregistration occurs	SAP Control Center.

Server Administration

The goal of server administration is to ensure that SAP Mobile Server is running correctly and that it is configured correctly for the environment in which it is installed (development or production). Server administration is mostly a one-time or infrequent administration task.

There are two types of server nodes that can be installed:

- Application Service node – (mandatory) runs all services.
- Scale Out node – (optional) specifically designed to allow the stateless request/response HTTP and synchronous message services to be horizontally scaled.

Table 2. Server administration tasks

Task	Frequency	Accomplished by
Installing the server	One-time installation per server	SAP Mobile Platform installer.
Reset JCo repository caches	Post installation	SAP Control Center
Manage the outbound enabler configuration for Relay Server. • Configure Relay Server properties • Manage certificates • View logs • Configure proxy servers for outbound enabler	Post installation	SAP Control Center

Application and User Management Overview

The goal of application management is to register an application to SAP Mobile Server as an entity, create an application template that specifies application connection details for a user, and activate application connections either manually or automatically.

Developers must invoke registration (manual or automatic) for native applications. For development details, see the *Developer Guide* for your application API and device platform type. For application, connection, and registration details, see *Applications* in the list of links below. .

Table 3. Application and user management tasks

Task	Frequency	Accomplish by using
Create new applications to register application entities with SAP Mobile Server. A default application template is created automatically. Modify and delete applications as part of application life cycle.	As required	SAP Control Center for SAP Mobile Platform with Applications node, and Applications tab.
Create or modify application connection templates to specify details for native, Hybrid App, and proxy application connections.	As required	SAP Control Center for SAP Mobile Platform, with Applications node, and Application Connection Templates tab.
Create one or more push configurations for applications.	As required	SAP Control Center for SAP Mobile Platform, with Applications node expanded and an application selected, and Push Configurations tab.
For applications that need to be registered manually, register an application connection to associate an application connection with a user. This is not necessary for applications that are registered automatically.	As required	SAP Control Center for SAP Mobile Platform, with Applications node, and Application Connections tab.
View activated users, once they have logged in with the activation code. Users must either supply the activation code manually, or the device client supplies the activation code automatically as coded.	As required	SAP Control Center for SAP Mobile Platform with the Applications node, and Application Users tab.
Create a new activation code for a user whose code has expired.	As required	SAP Control Center for SAP Mobile Platform, with Applications node, and Application Connections tab.
Review registered application connections and users, delete application connections to free licenses, delete application connections to remove users from the system.	As required	SAP Control Center for SAP Mobile Platform with the Applications node.
Change logical roles or modify role mappings for a security configuration to prevent users from accessing the application.	As required	SAP Control Center for SAP Mobile Platform with the Security node
Manage subscriptions	As required	SAP Control Center for SAP Mobile Platform with the Packages node.

Information and guidelines:

- Application templates hold default connection properties that can be assigned to an application during the connection registration process. However, these templates are configured differently depending on what type of connection registration you enable for applications. See the recommendations documented in *Creating Application Connection Templates* in the *Mobile Application Life Cycle* guide.

When a client application connects to SAP Mobile Server its application ID is used to look up a matching template. If that template allows automatic registration (the Automatic Registration Enabled property is set to true), then the security configuration in the template is used to authenticate the user and establish an identity against which the connection is registered. If the template also specifies a logical role, then user is authorized using the mapped physical role(s) of the logical role in the security configuration. When there are templates with different logical roles for the same application id and security configuration, the priority of the template determines the order of evaluation of the associated logical roles.

Note: If no templates are detected, the registration request fails. If multiple templates are detected, the client application registers using the template with the highest priority. If there is more than one template with the highest priority, the application registers using one of the templates with the highest priority, selected at random. For details on how user names and security configuration names are processed when an email address is used, see *Considerations for Using E-mail Addresses as User Names* in the *Security* guide.

See also

- *Applications* on page 253

Domain Management

The goal of domain management is to create and manage domains for one specific tenant. Use multiple domains for multiple tenants sharing the same SAP Mobile Server cluster.

Multiple domains in a cluster allow tenants' administrators (that is, domain administrators) to each manage their own application components. Domain administration for the platform administrator is typically an infrequent administration task that occurs each time a new domain needs to be added to support a change in the tenancy strategy used or need to make changes to an existing domain.

Domains give you the means to logically partitioning environments, thereby providing increased flexibility and granularity of control over domain-specific applications. Administration of multiple customer domains takes place within the same cluster.

- An platform administrator adds and configures domains, creates security configurations for customer applications, and assigns those security configurations to the domain so they can be mapped to packages in the domain.
- One or more domain administrators then perform domain-level actions within their assigned domains.

In a development environment, domains allow developers from different teams to share a single cluster without disrupting application deployment. Administrators can facilitate this by:

1. Creating a domain for each developer or developer group.
2. Granting domain administration privileges to those users so they can perform deployment tasks within their assigned domains.

Table 4. Domain management tasks

Task	Frequency	Administrator
Create domains	Once for each customer	SAP Mobile Platform administrator
Create and assign security configurations, and map roles at package or domain levels	Infrequent, as required	SAP Mobile Platform administrator
Assign and unassign domain administrators	Infrequent, as required	SAP Mobile Platform administrator
Configure and review domain logs	Routine	SAP Mobile Platform administrator and domain administrator
Deploy MBO and DOE-C packages	Routine	SAP Mobile Platform administrator and domain administrator
Manage server connections and templates	Infrequent, as required	SAP Mobile Platform administrator and domain administrator
Manage subscriptions and scheduled tasks	As required	SAP Mobile Platform administrator and domain administrator
Review client log and MBO/operation error history	As required	SAP Mobile Platform administrator and domain administrator

MBO Package Management Overview

The goal of mobile business object (MBO) package management is to make MBOs available to device users. MBO package management typically requires a one-time deployment and configuration, except for ongoing subscription management for messaging and SAP® Data Orchestration Engine Connector (DOE-C) packages.

Packages contain MBOs that are deployed to SAP Mobile Server to facilitate access to back-end data and transactions from mobile devices. Package types include UNIFIED packages, and SAP® DOE-C packages.

A package, along with its current settings for cache groups, role mappings, synchronization groups, connections, and security configuration, can be exported to an archive and imported back into SAP Control Center for backup or to facilitate a transition from a test environment to a production environment.

Table 5. MBO package management tasks

Task	Package type	Frequency	Accomplish by using
Deploy packages to a pre-production SAP Mobile Server	UNIFIED	Once, unless a new version becomes available	SAP Control Center for SAP Mobile Platform with the domain-level Packages node
Control user access by assigning security configurations for each package, and mapping roles if fine-grained authorization is enforced through logical roles	UNIFIED	Once, unless security requirements of the package change	SAP Control Center for SAP Mobile Platform with the domain-level Packages node
Set up the package cache interval and cache refresh schedule (for getting data updated on the SAP Mobile Server from the data source)	UNIFIED	Once, unless data refreshes need to be tuned	SAP Control Center for SAP Mobile Platform with the domain-level Packages node
Manage subscriptions (UNIFIED, and DOE-C), synchronization groups (UNIFIED), and device notifications (UNIFIED) to customize how updated data in the cache is delivered to the device user	Varies	Periodic, as required	SAP Control Center for SAP Mobile Platform with the domain-level Packages node
Export or import an MBO package to a QA or production SAP Mobile Server	UNIFIED	On-demand, as required	SAP Control Center for SAP Mobile Platform with the domain-level Packages node
Review current/historical/performance metrics	All	Routine	SAP Control Center for SAP Mobile Platform with the Monitoring node (available only to administrators)

Task	Package type	Frequency	Accomplish by using
View asynchronous operation replays for the selected package	Replication, UNIFIED	Periodic, as required	SAP Control Center for SAP Mobile Platform with the domain-level Packages node. However, asynchronous operation replays must first be enabled at the cluster level. See <i>Viewing Asynchronous Operation Replays</i> .

See also

- *Viewing Asynchronous Operation Replays* on page 356

Hybrid App Package Management Overview

The goal of Hybrid App package management is to make Hybrid Apps available from the SAP Mobile Server to device users. Hybrid App package management typically requires a one-time deployment and configuration, except for ongoing package maintenance.

The Hybrid App application is a simple business process application that delivers functionality, such as sending requests and approvals through an e-mail application, to mobile device clients on supported device platforms, including Windows Mobile, iOS.

Table 6. Hybrid App package management

Task	Frequency	Accomplish by using
Deploy Hybrid App packages	Once, unless a new version becomes available	SAP Control Center for SAP Mobile Platform with the Hybrid Apps node
Assign or unassign a Hybrid App to an application connection template	When a new Hybrid App package is deployed	SAP Control Center for SAP Mobile Platform with the Hybrid Apps node
Hybrid App configuration that includes e-mail matching rules and context variables	Once	SAP Control Center for SAP Mobile Platform with the Hybrid Apps node
Device registration and user assignments to Hybrid App packages	Routine when new users or new devices are added	SAP Control Center for SAP Mobile Platform with the Hybrid Apps > <Hybrid App-Name> node
Monitor users and errors	Routine	SAP Control Center for SAP Mobile Platform with the Monitoring node

Security Administration Overview

Perform security administration tasks to establish rules for the protection of enterprise and administrative data and transactions.

SAP Mobile Server coordinates data between enterprise information server (EIS) data sources and device clients, meaning that transferred information is often proprietary, confidential, or private. Therefore, the data and communication streams that carry information from SAP Mobile Server to other components in the SAP Mobile Platform must be protected.

SAP Mobile Platform has several security layers that protect data and transactions. Administrators manage system and application authentication and authorization security configurations at the cluster level, and perform role mapping at the domain and package levels. By default, the 'admin' security configuration is used to authenticate and authorize all administrative users, including domain administrators. All domain administrator logins must be valid in the security repository configured for the 'admin' security configuration.

Platform administrators register domain administrators at the cluster level, and then assign them to a domain from the domain-level Security Configurations tab. Security configurations are assigned when domains are created, or subsequently, from the Domains node. Packages must also be mapped to a security configuration at deployment; role mapping can be configured at a later time.

Roles are used for MBOs and operations during development to indicate authorization requirements. These roles are enforced by SAP Mobile Server. At deployment or after deployment, these logical roles can be mapped to physical roles to restrict which users have access to MBOs and operations. Roles assigned at the MBO level are separate from operation-level roles. However, package-level role mapping overrides domain-level role mapping. If the same package is deployed to multiple domains and associated with the same security configuration, then the domain-level role mapping is shared.

System Monitoring Overview

(Not applicable to Online Data Proxy) The goal of monitoring is to provide a record of activities and performance statistics for various elements of the application. Monitoring is an ongoing administration task.

Use monitoring information to identify errors in the system and resolve them appropriately. This data can also be shared by platform and domain administrators by exporting and saving the data to a .CSV or .XML file.

Note: Monitoring is only supported for object API applications. The equivalent support for OData applications is in domain logging. There are specific log buckets for OData based applications (Proxy Request/Response, Push, Error). To set up domain logging for OData, you configure a domain log in the default domain, create a domain log profile in a domain where application exists to specify the logging scope, and use SAP Control Center to trace the

activities. See *SAP Control Center for SAP Mobile Platform > Administer > Domains > Domain Logs*.

The platform administrator uses SAP Control Center to monitor various aspects of SAP Mobile Platform. Monitoring information includes current activity, historical activity, and general performance during a specified time period. You can monitor these components:

- Security log
- Replication synchronization
- Messaging synchronization
- System messaging queue status
- Data change notifications
- Device notifications (replication)
- Package statistics (replication and messaging)
- User-related activity
- Cache activity

To enable monitoring, platform administrators must set up a monitoring database, configure a monitoring data source or create a new one, and set up monitoring database flush and purge options. By default the installer created a monitoring database, however you can use another one if you choose. You must activate an application with a registered connection before you can monitor it.

To control monitoring, platform administrators create monitoring profiles and configurations, which define the targets (domains and packages) to monitor for a configured length of time. A default monitoring profile is created for you by the installer. Monitoring data can be deleted by the platform administrator as needed.

Table 7. System monitoring tasks

Task	Frequency	Accomplished by
Create and enable monitoring profiles	One-time initial configuration with infrequent tuning as required	SAP Control Center for SAP Mobile Platform with the Monitoring node
Enable domain logging	One-time setup with infrequent configuration changes, usually as issues arise	SAP Control Center for SAP Mobile Platform with the Domain node. Expand the node and select <domainName> > Log .
Review current/historical/performance metrics	Routine	SAP Control Center for SAP Mobile Platform with the Monitoring node
Identify performance issues	Active	SAP Control Center for SAP Mobile Platform with the Monitoring node

Task	Frequency	Accomplished by
Monitor application and user activity to check for irregularities	Active	SAP Control Center for SAP Mobile Platform with the Monitoring node
Troubleshoot irregularities	Infrequent	Reviewing various platform logs
Purge or export data	On demand	SAP Control Center for SAP Mobile Platform with the Monitoring node

Starting and Stopping SAP Control Center in SAP Mobile Platform

You can start and stop SAP Control Center through the Windows Services control panel or by using a command-line script.

Starting or stopping SAP Control Center through Windows Services

1. Open Windows Services by doing one of the following:
 - In Windows XP, open the Windows Control Panel and select **Administrative Tools > Services**.
 - In Windows 7, right click the My Computer icon on the desktop, then select **Manage > Services and Applications > Services**.
2. Locate SAP Control Center *X.X*. If the service is running, the status column displays “Started”.
3. To start or stop the service, right-click the service and choose **Start** or **Stop**.
4. Double-click the service.
5. To set the service to automatically start when the system starts, choose the **General** tab and change the **Startup type** to Automatic.
6. To restart the service in case of failover, choose the **Recovery** tab and change the First, Second, and Subsequent failures to Restart Service.
Click **Apply** to save the modifications before closing the dialog.

To start or stop SAP Control Center using a command-line script

- To start SAP Control Center, enter the start command: `SCC_HOME\bin\scc.bat`
- To stop SAP Control Center, enter the stop command: `SCC_HOME\bin\scc.bat - stop`

Note: You can use **scc.bat -stop** only to stop an SAP Control Center that was manually started with **scc.bat**; it cannot stop the SAP Control Center Windows service.

Cleaning Up the Flash Player Cache

SAP recommends you clean up the Flash Player cache, after upgrading to the latest version of SAP Control Center. This is needed if you have used a previous version of SAP Mobile Platform on the same machine. This cleanup is only required once.

1. Navigate to `C:\Documents and Settings\<username>\Application Data\Macromedia\Flash Player\#SharedObjects`.

Note: If you cannot access this folder, copy `C:\Documents and Settings\<username>\Application Data\Macromedia\Flash Player\#SharedObjects` to the address field and press **Enter**.

2. Delete all files under this folder.

Note: Alternatively, go to the following link from a browser: http://www.macromedia.com/support/documentation/en/flashplayer/help/settings_manager07.html. Use the Website Storage settings panel to change storage capacity, or delete Websites to clean up the cache.

You can access the tool from Adobe Flash Player online help (scroll to the "Website Storage Settings panel" link): <http://helpx.adobe.com/flash-player/kb/disable-local-shared-objects-flash.html>.

Copying and Pasting Properties

Values displayed in property tables in SAP Control Center can be copied and pasted.

Tables that support copying and pasting include monitoring properties, device properties, user properties, registration templates, domain log properties, and sever log properties.

1. To copy a value, right click the cell, then select **Copy** from the context menu.
2. To paste what you have copied, go to the property table you require, click the cell in question, then select **Paste** from the context menu. You cannot paste in a table cell that is read only, by you can copy a value from a table cell and paste it elsewhere (for example, copy text input for a search).

Getting Started with Production Clusters

Get started using SAP Control Center in production clusters of SAP Mobile Platform. Follow steps to configure and prepare SAP Control Center for SAP Mobile Platform use.

1. *Getting Started After Installing*

Perform postinstallation testing and configuration.

2. *Configuring Memory Usage*

(Optional) Determine whether you need to configure how much memory SAP Control Center uses, and if so which configuration method to use.

3. *Configuring the Automatic Logout Timer*

(Optional) Set SAP Control Center to end login sessions when users are inactive for too long.

4. *Manually Opening the SAP Mobile Platform Console*

If the SAP Mobile Platform administration console does not appear automatically, you may need to manually open it in SAP Control Center (SCC). Once open, you can then use the SAP Mobile Platform administration console to manage the SAP Mobile Server enabled mobile environment.

5. *Adding or Updating server-name Registration Properties* *SAP Mobile Server Registration Properties*

By default SAP Control Center automatically detects and registers clusters and SAP Mobile Server nodes as managed resources. The resource entry named 'localhost' is created for the local server upon installation. However, under certain conditions, you need to manually register new clusters or nodes, or modify existing entries.

Getting Started After Installing

Perform postinstallation testing and configuration.

Prerequisites

Start SAP Control Center.

Task

1. Install Adobe Flash Player 10.1 or later in the Web browser you will use to connect to SAP Control Center.

Flash Player is a free plug-in. You can download the latest version from <http://get.adobe.com/flashplayer/>.

If Flash Player is already installed but you are not sure which version you have, go to the Adobe test site at <http://adobe.com/shockwave/welcome>. Click the link that says **Test your Adobe Flash Player installation**. The version information box on the next page that appears displays your Flash Player version.

2. To connect to SAP Control Center, direct your browser to:

`https://<scc_server_hostname>:8283/scc`

Note: If you changed the default HTTPS port during installation, use the new port number instead of 8283.

3. If you see an error about the security certificate, add SAP Control Center to your browser's trusted sites zone (Internet Explorer) or add a security exception (Firefox).
4. Log in. Use the login account (supAdmin) and password that you set up during installation. This account can be used for both SAP Control Center login, and SAP Mobile Platform login.
5. Learn about SAP Control Center. To open the help system, click ? in the upper-right corner of the screen, or select **Help > Online Documentation**.

See also

- *Configuring Memory Usage* on page 17

Setting Up Browser Certificates for admin-tooling-name ConnectionsSAP Control Center Connections

To avoid security exceptions when launching SAP Control Center, set up security certificates correctly.

This task is required when:

- The browser session starts from a host computer that is remote from the SAP Control Center installation.
- The browser session starts on the same computer as SAP Control Center and reports a Certificate Error. The installer automatically sets up a local security certificate, but the certificate installed for https in the web container keystore is a self-signed root certificate, which is not recognized by the client browser.
- The host computer does not have Visual Studio Certificate Manager SDK installed.

Alternatively, follow browser-specific instructions to accept the certificate into the Windows certificate store.

1. Change the default shortcut to use the full host name of the computer on which SAP Control Center has been installed.

The host name is required because the default self-signed generated certificate the installer issues cannot be assigned to "localhost."

For example, change the shortcut URL to something similar to:

```
"%ProgramFiles%\Internet Explorer\iexplore.exe" https://SCCHost.mydomain.com:8283/scc
```

2. Add the certificate to the Windows certificates store.

- a) Extract the self-signed certificate:

```
SMP_HOME\sapjvm_7\bin\keytool.exe -exportcert -alias jetty  
-keystore SCC_HOME\services\EmbeddedWebContainer\keystore -  
file cert.crt
```

- b) Click **Start > Run**, type `mmc`, and then click **OK** to import the `cert.crt` file into the host computer's Windows store with the Windows Certificate Manager. The default password for both the keystore and the alias is "changeit".

Logging Into admin-tooling-name with an Installer-Defined PasswordSAP Control Center with an Installer-Defined Password

The person acting as platform administrator logs in to SAP Control Center for the first time after installation.

During installation, the person installing SAP Mobile Platform defines a password for the supAdmin user. This password is used to configure the Preconfigured login module that performs the administrator authentication.

Note: This installer-defined password is not intended to be a permanent administrator credential. You must replace this module with a production-grade authentication module, typically LDAP.

1. Launch SAP Control Center.
2. Enter supAdmin for the user name and type the <supAdminPwd> for the password.
Note that the user name is case sensitive.
3. Click **Log in**.

Logging out of admin-tooling-nameSAP Control Center

Log out of a cluster when you finish your administration session.

In order to protect system security, SAP recommends that you log out of SAP Control Center when you are not using the console.

Choose one of these methods:

- Click the **Log out** link at the top right corner of the console.
- From the SAP Control Center menu, select **Application > Log out**.

Configuring Memory Usage

(Optional) Determine whether you need to configure how much memory SAP Control Center uses, and if so which configuration method to use.

It is not usually necessary to configure memory usage for SAP Control Center. This table lists memory options you can set and circumstances under which you should consider changing them.

Modify this value	When	Guidelines
Maximum memory <code>jvmopt=-Xmx</code> – if you are running SAP Control Center as a Windows service <code>SCC_MEM_MAX</code> – if you are starting SCC from the command line	- You need to prevent SAP Control Center from using more than a given amount of memory - SAP Control Center fails to start and may display an error: <code>Could not create the Java Virtual machine.</code> - An <code>OutOfMemory</code> error says SAP Control Center is out of heap space - A warning message about system memory appears during the start process - The machine where SAP Control Center is installed has less than 4GB of memory. (Starting SAP Control Center on a machine with less than 4GB of memory triggers the startup warning message about system memory.)	On machines with less than 4GB of memory, set maximum memory to 256MB or more. Default value: none. (On machines with 4GB or more of memory, maximum memory is set dynamically and is effectively limited only by the amount of system memory available.)
Permanent memory <code>jvmopt=-XX:MaxPermSize</code> – if you are running SAP Control Center as a Windows service <code>SCC_MEM_PERM</code> – if you are starting SCC from the command line	An <code>OutOfMemory</code> error says SAP Control Center is out of permanent generation space	Increase by 32MB increments. If you reach a value equal to twice the default and still see the <code>OutOfMemory</code> error, contact SAP technical support. Default value: 128MB

You can change memory options in two ways:

- For SAP Control Center started from the command line – execute commands to set one or more environment variables before executing the **scc** command to start SAP Control Center. When you use this method, your changes to the memory options last only as long as the current login session. This method is useful for testing new option values.

- For the SAP Control Center service – modify a file used by the SAP Control Center service. When you use this method, your changes to the memory options persist—SAP Control Center uses them every time it starts as a service.

See also

- *Getting Started After Installing* on page 15
- *Configuring the Automatic Logout Timer* on page 20

Changing a Memory Option on the Command Line

Before you start SAP Control Center from the command line, you can issue a command to change the value of a memory option temporarily.

Changes made using this method last only as long as the current login session. This method is useful for testing new option values.

1. If SAP Control Center is running, shut it down.
2. Set the environment variable. Specify a size in megabytes but do not indicate the units in the command.

```
> set SCC_MEM_MAX=512
```

3. Use the **scc** command to start SAP Control Center.

Changing a Memory Option for a SAP Control Center Windows Service

Add a **jvmopt** command to the `scc.properties` file to change a memory option (`-Xmx` or `-XX:MaxPermSize`) for a SAP Control Center Windows service.

When you use this method to set memory options, your changes are permanent—SAP Control Center uses them every time it starts as a service.

1. If SAP Control Center is running, shut it down.
2. Open the SAP Control Center properties file:

```
<SAP Control Center-install-directory>\SCC-3_2\bin  
\scc.properties
```
3. Add (or modify, if it already exists) a **jvmopt** line specifying the memory size in Java format. Use `m` for megabytes or `g` for gigabytes.

For example:

```
jvmopt=-Xmx512m
```

4. Save the file and start the SAP Control Center Windows service.

Configuring the Automatic Logout Timer

(Optional) Set SAP Control Center to end login sessions when users are inactive for too long.

Prerequisites

Launch SAP Control Center and log in using an account with administrative privileges. (The login account or its group must have sccAdminRole.)

Task

1. From the application's menu bar, select **Application > Administration**.
2. Select **General Settings**.
3. Click the **Auto-Logout** tab.
4. Enter the number of minutes after which an idle user will be automatically logged out.
Enter 0 or leave the box empty to disable automatic logout.
5. Click **OK** (to apply the change and close the properties dialog) or **Apply** (to apply the change and leave the dialog open).

See also

- *Configuring Memory Usage* on page 17

Manually Opening the SAP Mobile Platform Console

If the SAP Mobile Platform administration console does not appear automatically, you may need to manually open it in SAP Control Center (SCC). Once open, you can then use the SAP Mobile Platform administration console to manage the SAP Mobile Server enabled mobile environment.

Prerequisites

Before managing a cluster, ensure that the login has SAP Control Center administration privileges.

Task

1. In the SAP Control Center menu, select **View > Open > Resource Explorer**.
2. From the list of resources, select the cluster you want to manage.
3. From the Resource Explorer menu bar, click **Resources > Add Resources to Perspective**.
The SAP Mobile Server is added to the Perspective Resources window.
4. In the Perspective Resources window, mouse over the cluster you want to manage, click the down arrow, and select **Authenticate**.

5. To authenticate against the cluster, select one of these:

- **Use my current SCC login** – SCC uses the administrator's initial SCC login credentials to establish a connection to the SAP Mobile Platform cluster. Use this option if you have already mapped the SAP Control Center administrator role to the SUP administrator role.
- **Specify different credentials** – enter a new user name and password specifically for logging in to this cluster. Use this option if SAP Control Center and SAP Mobile Platform use different authentication repositories. Using different credentials in this step is unnecessary if SAP Control Center and SAP Mobile Platform use the same security provider.

6. Click **OK**.

7. Mouse over the cluster you want to open, click the down arrow, and select **Manage**.

If you are successfully authenticated, the SAP Mobile Platform console appears. If authentication fails, see *Troubleshoot SAP Control Center* in the *Troubleshooting* guide.

Adding or Updating server-name Registration PropertiesSAP Mobile Server Registration Properties

By default SAP Control Center automatically detects and registers clusters and SAP Mobile Server nodes as managed resources. The resource entry named 'localhost' is created for the local server upon installation. However, under certain conditions, you need to manually register new clusters or nodes, or modify existing entries.

Prerequisites

You must manually create a new registration entry when:

- a cluster or node is not located within your network.
- a cluster or node is not automatically detected and registered in the SAP Control Center Resource Explorer.

You must manually update an existing registration entry when you modify the management port for the SAP Mobile Server configuration. You must update the resource properties to match the new values.

Note: When modifying the hostname of the resource, you need to reauthenticate the resource.

Task

1. Choose your action:

- To register a new resource, on the SAP Control Center menu, select **Resource > Register**.

- To update the resource properties, on the SAP Control Center menu, select **View > Select > Perspective Resources** view. Then in the Name column, click **EntryName > Properties**.
2. Configure any of these properties, depending on you initial action:
 - the resource name and type
 - a description
 - host name and port of the server
The host name and port must match those configured for the SAP Mobile Server management port.
 - Use Secure Connection - set this property to **Yes** if using a secure port; set to **No** if using a non-secure port.
 3. If you changed hostname, reauthenticate the server:
 - a) Click **EntryName > Clear Authentication** to remove currently validated credentials to the previous host values.
 - b) Click **EntryName > Authenticate** to reauthenticate with the current host values.
 4. Once authenticated, you can now manage it from SAP Control Center: click **EntryName > Manage** to launch the SAP Mobile Platform management console.

See also

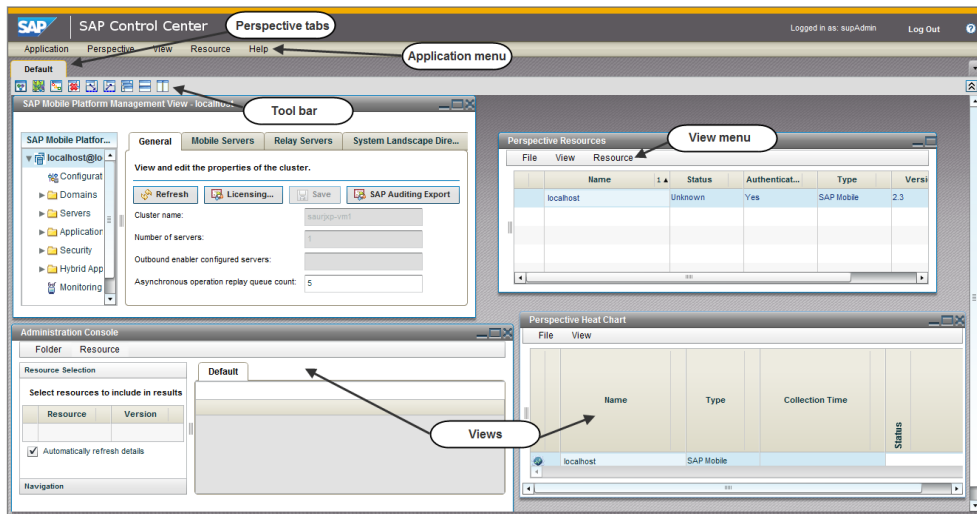
- *Configuring Management Port Properties* on page 42

Understanding the SAP Control Center Interface

Manipulate SAP Control Center interface elements to set up the console according to your requirements and preference.

User Interface Overview

This illustration labels important elements of the SAP Control Center user interface so you can identify them when they appear in other help topics.

Figure 1: SAP Control Center User Interface**See also**

- *Perspectives* on page 24
- *Views* on page 26
- *Repository* on page 28
- *SAP Control Center Console* on page 33
- *SAP Control Center Security* on page 36

Toolbar Icons

Describes the icons in the SAP Control Center toolbar for launching and managing views.

Table 8. Toolbar Icons

Icon	Name	Description
	Show/Hide Perspective Resources View	Displays or minimizes the Perspective Resources view, which lists registered resources in this perspective.
	Launch Resource Explorer	Opens the resource explorer, which lists reachable resources (both registered and unregistered).
	Launch Heat Chart	Opens the perspective heat chart, which gives a status overview of the registered resources in this perspective.

Icon	Name	Description
	Close All Open Views	Closes all open and minimized views.
	Minimize All Open Views	Minimizes all open views.
	Restore All Minimized Views	Returns all minimized views to their original size.
	Cascade All Open Views	Arranges open views to overlap each other.
	Tile All Open Views Vertically	Arranges open views in a vertical manner.
	Tile All Open Views Horizontally	Arranges open views in a horizontal manner.

SAP Control Center Functionality Not Applicable to SAP Mobile Platform

SAP Control Center is a standard management framework used by multiple products, including SAP Mobile Platform. Certain standard functions that appear in the user interface cannot be used to administer SAP Mobile Platform.

The following SAP Control Center features can be disregarded in the context of SAP Mobile Platform:

- Alerts
- Schedules
- Heat charts
- Historical performance monitoring

These features either do not apply to SAP Mobile Platform or are redundant due to custom functionality implemented in place of standard functions. The inapplicable SAP Control Center functionality cannot be removed, as it may be required by other SAP product servers also using SAP Control Center.

Perspectives

A perspective is a named container for a set of one or more managed resources. You can customize perspectives to provide the information you need about your environment.

As the main workspaces in the SAP Control Center window, perspectives let you organize managed resources. You might assign resources to perspectives based on where the resources are located (continents, states, or time zones, for example), what they are used for, which group owns them, or which administrator manages them. Perspectives appear as tabs in the main window.

Every perspective includes a Perspective Resources view, which lists the resources in that perspective and provides high-level status and descriptive information. Use the View menu to switch from detail view to icon view and back.

You can open additional views as needed to manage the perspective's resources. The views in a perspective display information only about resources in that perspective.

One resource can appear in many perspectives.

See also

- *User Interface Overview* on page 22
- *Views* on page 26
- *Repository* on page 28
- *SAP Control Center Console* on page 33
- *SAP Control Center Security* on page 36

Creating a Perspective

Create a perspective in which you can add and manage resources.

1. From the application menu bar, select **Perspective > Create**.
2. Enter a name for your perspective. The name can contain up to 255 characters.
3. Click **OK**.

See also

- *Removing a Perspective* on page 25
- *Renaming a Perspective* on page 25

Removing a Perspective

Delete a perspective window.

1. Select the perspective tab you want to delete.
2. In the main menu bar, select **Perspective > Delete**.
The selected perspective disappears. If there are other perspectives, SAP Control Center displays one.

See also

- *Creating a Perspective* on page 25
- *Renaming a Perspective* on page 25

Renaming a Perspective

Change the name of your perspective.

1. Select the perspective tab you want to rename.

2. From the main menu bar, select **Perspective > Rename..**
3. Enter the new name for your perspective.
4. Click **OK**.

See also

- *Creating a Perspective* on page 25
- *Removing a Perspective* on page 25

Views

Use views to manage one or more resources within a perspective.

In SAP Control Center, views are the windows you use to monitor and manage a perspective's resources. You can re-arrange, tile, cascade, minimize, maximize, and generally control the display of the views in your perspective.

Each perspective includes these views:

- Perspective Resources
- Administration Console

Note: SAP Control Center views are not related to database views; they serve a completely different purpose.

See also

- *User Interface Overview* on page 22
- *Perspectives* on page 24
- *Repository* on page 28
- *SAP Control Center Console* on page 33
- *SAP Control Center Security* on page 36

Managing a View

Open, close, minimize, maximize, or restore a view in the current perspective.

You can:

Task	Action
Open a view	Do one of the following: • In the Perspective Resources view, select a resource, click the drop-down arrow to the right of the resource name, and select the view to open. • In the application menu bar, select View > Open and choose a view.

Task	Action
Close a view	Select the view to close. In the application menu bar, select View > Close . You can also click the X in the view's upper right corner.
Maximize a view	Click the box in the view's upper right corner. The view enlarges to fill the entire perspective window. Click the box again to return the view to its former size.
Minimize a view	Click the _ in the view's upper right corner. The view shrinks to a small tab at the bottom of the perspective window.
Minimize all views	In the application menu bar, select Perspective > Arrange > Minimize All Views .
Restore a view	Click the box on the minimized tab to maximize the view. Click the box again to return the view to its former (smaller) size so you can see other views at the same time.
Bring a view to the front	In the application menu bar, select View > Select and choose the view you want from the submenu.

See also

- *Arranging View Layout in a Perspective* on page 27

Arranging View Layout in a Perspective

Use the view layout options to manage your perspective space.

Click one of these icons in the SAP Control Center toolbar:

Icon	Action
	Close All Open Views
	Minimize All Open Views
	Restore All Minimized Views
	Cascade All Open Views
	Tile All Open Views Vertically
	Tile All Open Views Horizontally

In a cascade, views overlap; in tiling arrangements, they do not.

Alternatively, you can arrange view layouts from the SAP Control Center menu bar. From the menu bar, select **Perspective > Arrange** and select your view layout.

See also

- *Managing a View* on page 26

Repository

The SAP Control Center embedded repository stores information related to managed resources, as well as user preference data, operational data, and statistics.

You can back up the repository database on demand, schedule automatic backups, restore the repository from backups, and configure repository purging options. Full and incremental backups are available. A full backup copies the entire repository. An incremental backup copies the transaction log, capturing any changes since the last full or incremental backup.

By default, SAP Control Center saves backups as follows:

- Each full backup is stored in its own subdirectory in `SCC_HOME/backup`.
- Each incremental backup is stored in a file in `SCC_HOME/backup/incremental`.

SAP recommends that you periodically move backup files to a secondary storage location to prevent the installation directory from becoming too large.

See also

- *User Interface Overview* on page 22
- *Perspectives* on page 24
- *Views* on page 26
- *SAP Control Center Console* on page 33
- *SAP Control Center Security* on page 36

Scheduling Backups of the Repository

Configure full and incremental backups of the repository to occur automatically.

Prerequisites

Determine your backup strategy, including when to perform full backups and incremental backups. For example, you might schedule incremental backups every day and a full backup every Saturday.

You must have administrative privileges (`sccAdminRole`) to perform this task.

Task

A full backup copies the entire repository. An incremental backup copies the transaction log, capturing any changes since the last full or incremental backup.

1. From the main menu, select **Application > Administration**.
2. In the left pane, select **Repository**.
3. Click the **Full Backup** tab.
4. (Optional) To change the directory in which backups will be stored, click **Browse** and navigate to the desired directory.
5. Select **Schedule a Regular Backup**.
6. Specify the day you want scheduled backups to begin. Enter a **Start date** or click the calendar and select a date.
7. (Optional) Use the **Time** and **AM/PM** controls to specify the time at which backups occur.
8. Specify how often backups occur by setting the **Repeat interval** and selecting hours, days, or weeks.
9. (Optional) To purge the repository after each backup, select **Run a repository purge after the backup completes**.
10. If you include purging in the backup schedule, go to the **Size Management** tab and unselect **Automatically purge the repository periodically** to disable automatic purging.
11. Click **Apply** to save the schedule.
12. Click the **Incremental Backup** tab and repeat the steps above to schedule incremental backups to occur between full backups.

Next

Set purging options on the Size Management tab.

See also

- *Modifying the Backup Schedule* on page 29
- *Forcing an Immediate Backup* on page 30
- *Restoring the Repository from Backups* on page 31
- *Configuring Repository Purging* on page 32

Modifying the Backup Schedule

Suspend or resume repository backups or change the backup schedule.

Prerequisites

You must have administrative privileges (sccAdminRole) to perform this task.

Task

1. From the main menu, select **Application > Administration**.
2. In the left pane, select **Repository**.
3. Choose the type of backup to modify:

- Click the **Full Backup** tab, or
 - Click the **Incremental Backup** tab.
4. (Optional) To suspend or resume the backup schedule, select or unselect **Schedule a Regular Backup**.
When you unselect (uncheck) this option, the scheduling area is grayed out and scheduled backups no longer occur. However, the schedule is preserved and you can reinstate it at any time.
 5. To change the backup schedule, edit the **Start date, Time, Repeat interval**, or units. You can also select or unselect **Run a repository purge after the backup completes**.
 6. Click **Apply** to save the schedule.

See also

- *Scheduling Backups of the Repository* on page 28
- *Forcing an Immediate Backup* on page 30
- *Restoring the Repository from Backups* on page 31
- *Configuring Repository Purging* on page 32

Forcing an Immediate Backup

Perform an unscheduled full or incremental backup of the repository.

Prerequisites

You must have administrative privileges (sccAdminRole) to perform this task.

Task

1. From the main menu, select **Application > Administration**.
2. In the left pane, select **Repository**.
3. Choose the type of backup to run:
 - Click the **Full Backup** tab, or
 - Click the **Incremental Backup** tab.
4. Click **Back up Now**.

SAP Control Center saves the backup to the directory shown in the Location field.

See also

- *Scheduling Backups of the Repository* on page 28
- *Modifying the Backup Schedule* on page 29
- *Restoring the Repository from Backups* on page 31
- *Configuring Repository Purging* on page 32

Restoring the Repository from Backups

Load backup files into the repository database to revert undesirable changes or to recover from a catastrophic failure.

If you configured SAP Control Center to store backups somewhere other than the default location, change the source directory in the copy commands in this procedure.

1. Shut down SAP Control Center.
2. Copy the most recent full backup from `SCC_HOME/backup/`
<generated_directory_name> to `SCC_HOME/services/Repository`.

For example:

```
copy C:\SCC_HOME\backup\repository.  
270110161105\scc_repository.db  
C:\SCC_HOME\services\Repository
```

3. If you have no incremental backups to load,
 - a) Also copy the log file from `SCC_HOME/backup/`
<generated_directory_name> to `SCC_HOME/services/Repository`. For example:

```
copy C:\SCC_HOME\backup\repository.  
270110161105\scc_repository.log  
C:\SCC_HOME\services\Repository
```

- b) Skip to step 5 on page 31.

4. (Optional) To load incremental backups, start the repository database using the **-ad** option, which directs it to load transaction logs (incremental backups) from the incremental directory. (The database loads full backups automatically.) For example:

```
cd SCC_HOME\services\Repository  
  
..\..\bin\sa\bin_<platform>\dbsrv11.exe scc_repository -ad  
SCC_HOME\backup\incremental
```

The repository database loads the full backup and any subsequent incremental backups present in the incremental directory. Incremental backups are loaded in date order. After loading and saving, the database shuts down.

5. Start SAP Control Center.
If you loaded incremental backups, SAP Control Center starts normally (that is, no further recovery occurs). If you copied a full backup to the `Repository` directory, the database recovers the repository from the full backup.

Example: Loading incremental backups into the repository database

These commands start SQL Anywhere® on a 32-bit Windows machine:

```
% cd C:\SCC_HOME\services\Repository  
% ..\..\bin\sa\bin_windows32\dbsrv11.exe scc_repository -ad  
C:\SCC_HOME\backup\incremental
```

See also

- *Scheduling Backups of the Repository* on page 28
- *Modifying the Backup Schedule* on page 29
- *Forcing an Immediate Backup* on page 30
- *Configuring Repository Purging* on page 32

Configuring Repository Purging

Change repository purging options.

Prerequisites

You must have administrative privileges (sccAdminRole) to perform this task.

Task

As you decide how to purge your repository, consider that:

- Purging keeps the repository from absorbing too much disk space.
- By default, purging is enabled. It occurs once a day and purges data older than one day.
- Statistics and alert history can help you detect trends in server performance and user behavior. The SAP Control Center statistics chart can graph performance data over a period of a year or more if the data is available. If you have enough disk space, consider saving data for a longer period of time or disabling the purging of statistics or alert history.
- Changing the purge frequency and other options might affect SAP Control Center performance.

Note: If you configure purging as part of a scheduled backup of the repository, disable automatic purging on the Size Management tab.

1. From the main menu bar, select **Application > Administration**.
2. Select **Repository**.
3. Click the **Size Management** tab.
4. To turn automatic purging on or off, click **Automatically purge the repository periodically**.
Turn this option off if purging is configured as part of your scheduled full or incremental backups.
5. Click purge options to turn them on or off:
 - **Purge statistics**
 - **Purge alert history**
6. In **Purge data older than**, enter the number of days after which to purge repository data.
7. Click **Apply**, then **OK**.

See also

- *Scheduling Backups of the Repository* on page 28
- *Modifying the Backup Schedule* on page 29
- *Forcing an Immediate Backup* on page 30
- *Restoring the Repository from Backups* on page 31

SAP Control Center Console

The console is a command-line interface for displaying details about the status of the SAP Control Center server and its subsystems.

When you use the **scc** command to start SAP Control Center, it displays start-up messages and then displays the console prompt.

Note: The console prompt does not appear if you start SAP Control Center as a service, if you direct the output of **scc** to a file, or if you start SAP Control Center in the background.

See also

- *User Interface Overview* on page 22
- *Perspectives* on page 24
- *Views* on page 26
- *Repository* on page 28
- *SAP Control Center Security* on page 36

Console Commands

Use the SAP Control Center console to get status information on SAP Control Center and its ports, plug-ins, and services.

help Command

Display syntax information for one or more SAP Control Center console commands.

Syntax

```
help [command_name]
```

Parameters

- **command_name** – optional. status, info, or shutdown. If you omit *command_name*, **help** returns information on all the console commands.

Examples

- **Example 1** – returns information on the **status** command:

```
help status
```

Permissions

help permission defaults to all users. No permission is required to use it.

See also

- *info Command* on page 34
- *shutdown command* on page 35
- *status Command* on page 35

info Command

Display information about specified parts of the SAP Control Center server.

If you enter **info** with no parameters, it returns information for every parameter.

Syntax

```
info [-a | --sys]
[-D | --sysprop [system-property]]
[-e | --env [environment-variable]]
[-h | --help]
[-m | --mem]
[-p | --ports]
[-s | --services]
```

Parameters

- **-a | --sys** – optional. List all the services known to SAP Control Center, indicate whether each service is enabled, and list other services on which each service depends.
- **-D | --sysprop [system-property]** – optional. Display information about the specified Java system property. Omit the system-property argument to return a list of all Java system properties and their values.
- **-e | --env [environment-variable]** – optional. List all the environment variables in the SAP Control Center Java VM process environment. Omit the environment-variable argument to return a list of environment variables and their values.
- **-h | --help** – optional. Display information about the **info** command.
- **-m | --mem** – optional. Display information about the server's memory resources.
- **-p | --ports** – optional. List all the ports on which the SAP Control Center agent and its services listen, indicate whether each port is in use, and show the service running on each port.
- **-s | --services** – optional. List all SAP Control Center services, indicate whether each service is enabled, and list other services on which each service depends.

Examples

- **Example 1** – displays information about ports on this SAP Control Center server:

```
info -p
```

Permissions

info permission defaults to all users. No permission is required to use it.

See also

- *help Command* on page 33
- *shutdown command* on page 35
- *status Command* on page 35

shutdown command

Stop the SAP Control Center server if it is running.

Syntax

```
shutdown
```

Examples

- **Example 1** – shuts down SAP Control Center:

```
shutdown
```

Permissions

shutdown permission defaults to all users. No permission is required to use it.

See also

- *help Command* on page 33
- *info Command* on page 34
- *status Command* on page 35

status Command

Display the status of the SAP Control Center agent, plug-in, or service components of SAP Control Center.

Syntax

```
status [-a | --agent]
[-h | --help]
[-p | --plugin [plugin-name]]
[-s | --service [service-name]]
```

Parameters

- **-a** | **--agent** – display the status of the SAP Control Center agent component.
- **-h** | **--help** – display information about the **info** command.
- **-p** | **--plugin** [**plugin-name**] – display the status of the specified SAP Control Center plug-in (for example, ASEMap, the Adaptive Server® management module). Omit the plugin-name argument to return a list of plug-ins.
- **-s** | **--service** [**service-name**] – display the status of the specified SAP Control Center service (for example, the Alert service or the Messaging service). Omit the service-name argument to return a list of services.

Examples

- **Example 1** – displays status information on the Repository service:

```
status --service RepositoryService
```

Permissions

status permission defaults to all users. No permission is required to use it.

See also

- *help Command* on page 33
- *info Command* on page 34
- *shutdown command* on page 35

SAP Control Center Security

User access to SAP Control Center is controlled by configuring a security provider. Security providers are configured with the SAP Mobile Platform management console.

By default, SAP Control Center delegates user access control to providers configured for SAP Mobile Server. Consequently, the login and group management features for SAP Control Center (that is, those available when you click **Application** > **Administration** > **Security** from the SAP Control Center menu) do not apply to the SAP Mobile Platform use case. See *Securing Platform Administration* in the *Security* guide.

See also

- *User Interface Overview* on page 22
- *Perspectives* on page 24
- *Views* on page 26
- *Repository* on page 28
- *SAP Control Center Console* on page 33

Platform Administration Roles and Tasks

By default, there are three logical administration roles for SAP Mobile Platform, each with a specific set of tasks to perform.

To enable role-based access to the SAP Control Center interface, configure mapping of the logical roles to roles that exist in the security repository used for administrative authentication and authorization. For details, see *Security*.

SAP Mobile Platform Administrator

Platform administrators interact with SAP Mobile Platform to perform high-level, cluster-wide management.

A platform administrator can perform all administrative operations in the SAP Mobile Platform administration console, including domain administration for all domains.

Note: The terms "SAP Mobile Platform (platform) administrator" is used in all documentation to refer to the user with "SUP Administrator" role.

Platform Administration Tasks

Review the tasks an SAP Mobile Platform administrator can perform.

Component	Available Tasks
Cluster	• Configure server properties for replication, messaging, management and HTTP ports, SSL, and performance. • View and manage server log settings. • Configure Relay Server and server farms. • Configure security configuration role mappings. • Review licensing status and generate SAP audit data. • Review and manage system landscape directory configuration, and upload schedule.
Domains	• Deploy, configure, and manage MBO packages. • Deploy and manage server connections and templates. • Configure security configuration role mappings. • Review and manage domain logs, domain log profiles, and log filters. Export domain log data. • Configure cluster-wide configuration objects in the "default" domain. These objects include the system database server connections and domain log. • View and manage domains. • Assign or unassign domain administrators to or from the domain.

Component	Available Tasks
Server	• Perform server starts and stops. • Configure outbound enablers on each node.
Security	• Create security configurations. • Specify authentication, authorization, and audit providers. • Configure authentication cache timeout and lockout properties. • Map logical roles to physical roles for security providers. • Assign named security configurations to domains or remove pre existing configurations where necessary. • Register new domain administrators.
Packages	• Import, export, deploy, and delete MBO packages. • Manage or view the properties of each package. For example, perform role mapping, configure security configuration and role mapping, manage subscriptions, and review and manage client log and MBO/ operation error history. • Enable or disable packages.
Hybrid Apps	• Deploy and configure Hybrid Apps. • Assign and unassign Hybrid Apps to users. • Review queue status and errors. • Import and export Hybrid Apps.
Applications	Import and export applications.
Monitoring	• Configure monitoring settings and review monitoring data. • Review captured monitoring data and KPIs .

SAP Mobile Platform Domain Administrator

Domain administrators interact with SAP Mobile Platform to manage packages, server connections, security configurations, and role mappings in specific domains. A domain is a logical partition used to isolate and manage runtime artifacts for a particular tenant.

The domain administrator can administer only the domain to which he or she is assigned. The domain administrator is granted access on a per-domain basis by the platform administrator.

The logical role for the domain administrator is "SUP Domain Administrator." The term "domain administrator" is used in all documentation to refer to the user with the "SUP Domain Administrator" role.

Domain Administration Tasks

Review the tasks a domain administrator can perform.

Component	Available Tasks
Domain	• Manage the domain. • Enable or disable managed domain. • Deploy, configure, and manage MBO packages. • Deploy and manage server connections and templates. • Configure security configuration role mapping. • Review and manage domain logs, domain log profiles, and log filters. Export domain log data. • Configure cluster-wide configuration objects in the "default" domain. These objects include the system database server connections and domain log.
Log	• Review and manage domain logs, domain log profiles, and log filters. • Export domain log data.
Packages	• Import, export, deploy, and delete MBO packages. • Manage or view the properties of each package, for example perform role mapping, configure security configuration and role mapping, manage subscriptions, review and manage client log and MBO/operation error history. • Enable or disable packages.
Connections	• Create and configure server connections. • Manage connection templates.
Security	• Map logical roles to physical roles for domain security providers.
Applications	• Review applications that are defined for the domain. • Review application connections that access the domain.

SAP Mobile Platform Helpdesk

Help desk operators interact with SAP Mobile Platform to review system information to determine the root cause of reported problems. Help desk operators only need to view information, not change it.

Help desk operators have read-only access to all administration information in the SAP Mobile Platform administration console. They cannot perform any modification operations on administration console tabs, and cannot save changes made in dialogs or wizards.

The logical role for the help desk operator is "SUP Helpdesk." The term "help desk operator" is used in all documentation to refer to the user with the "SUP Helpdesk" role.

Help Desk Operator Tasks

Review the tasks a help desk operator can perform.

Component	Available Tasks
Cluster	• Review properties for replication, messaging, management and HTTP ports, SSL, and performance. • Review server logs. • Review Relay Server and server farms. • Review licensing status and generate SAP audit data. • Review system landscape directory configuration and schedule.
Domain	• Review existing domains. • Review MBO packages and their configuration. • Review server connections and templates properties. • Review security configurations and role mapping. • Review domain logs, domain log profiles, and log filters. Export domain log data .
Server	• Review outbound enablers on each node.
Applications	• Review applications, application push configurations, application users and connections, and application connection templates.
Security	• Review security configurations, their authentication, authorization, attribution, and audit provider properties, and their logical role mappings. • Review authentication cache timeout and lockout properties. • Review security configurations assigned to. • Review currently registered domain.
Hybrid Apps	• Review currently deployed Hybrid Apps, their configuration settings, assigned application connections, application connection templates, queue status, and errors.
Monitoring	• Review monitoring settings and monitoring data. • Review captured monitoring data and KPIs.

Administer

Use SAP Control Center for SAP Mobile Platform to administer and configure components of a cluster registered as a managed resource. When you configure cluster components you are setting up the elements required to mobilize your data. Once configured you perform ongoing administration tasks to maintain the environment.

Clusters

As an organization grows, SAP Mobile Platform administrators need to create a scalable IT infrastructure using clusters. Clustering creates redundant SAP Mobile Platform components on your network to provide a highly scalable and available system architecture.

Organizations can seamlessly achieve high availability and scalability by adding more or redundant instances of core components. Redundant instances of critical components provide transparent failover.

In a production environment, the SAP Mobile Platform deployment typically uses at least one relay server. The connections to relay servers can be configured within a cluster instance from SAP Control Center.

Cluster-Affecting Configuration Changes

Before you configure SAP Mobile Servers in a cluster, ensure you understand how changes are synchronized to cluster members.

When you make a cluster-affecting change on the primary SAP Mobile Server, those changes are synchronized to all secondary servers in the cluster. This ensures that servers are configured the same way and behave consistently within the cluster.

Cluster-affecting changes include:

- cluster configuration
- monitoring setup
- security configuration

Cluster Properties

Cluster properties let administrators manage server configuration settings for all nodes in the cluster to ensure smooth data exchange between the server and client.

Cluster property changes that do not require a server restart will be automatically synchronized to all nodes in the cluster. A message is displayed for any changes that require a server restart. Servers that require a restart are also flagged in the Server List. Perform a restart using SAP Control Center or the Desktop shortcut (by stopping, then starting), as restarting from Windows services does not guarantee the changes taking effect.

Note: To view or modify properties on the Component or Performance tabs, wait for the refresh of cluster configuration data to complete. Otherwise, your updates cannot take effect.

See also

- *Server List* on page 85
- *SAP Mobile Server* on page 84
- *Server Log* on page 96

Configuring Management Port Properties

Management ports in SAP Mobile Server process incoming administration connection requests from SAP Control Center. Management ports use IIOPS by default, though IIOP can be configured as well.

Note: If you are using SAP Control Center in a development/test environment, you must also configure the SAP Mobile WorkSpace on all development computers accessing the development SAP Mobile Server to also save the required certificates to the java keystore. SAPMutual authentication for the IIOPS management port is not supported.

1. In the left navigation pane, select **Configuration**.
2. In the right administration pane, click the **General** tab.
3. From the menu bar, select **Components**.
4. Select **Management**, then click **Properties**.
5. To configure the IIOP management port, enter the port number (default is 2000).
6. To configure the IIOPS secure management port:
 - Enter the port number (default is 2001).
 - Select the security profile used for the secure port.

See also

- *Replication* on page 43
- *Messaging* on page 45
- *Enforcing OData Application Registration* on page 46
- *Upgrade: Configuring Data Change Notification Components* on page 47
- *Configuring Cluster Performance Properties* on page 47
- *Configuring SSL Properties* on page 51
- *Configuring Web Container Properties* on page 56
- *Configuring Cache Properties* on page 58
- *Configuring Server Log Settings* on page 59
- *Adding or Updating server-name Registration Properties* *SAP Mobile Server Registration Properties* on page 21

Replication

Replication synchronization involves synchronization between SAP Mobile Server and a replication-based mobile device application. Synchronization keeps multiple variations of the data set used by a device application in coherence with one another by reconciling differences in each. Reconciling differences before writing updates back to the enterprise information server (EIS) maintains data integrity.

For replication synchronization, configure the corresponding port to receive incoming synchronization requests from devices, as well as set up configuration to enable push notification messages to the device when data changes in CDB. In a typical environment, client applications running on devices will connect to the synchronization port via Relay Server and Relay Server Outbound Enabler (RSOE). In those cases, the HTTP port will be used.

See also

- *Configuring Management Port Properties* on page 42
- *Messaging* on page 45
- *Enforcing OData Application Registration* on page 46
- *Upgrade: Configuring Data Change Notification Components* on page 47
- *Configuring Cluster Performance Properties* on page 47
- *Configuring SSL Properties* on page 51
- *Configuring Web Container Properties* on page 56
- *Configuring Cache Properties* on page 58
- *Configuring Server Log Settings* on page 59

Configuring a Replication Listener

(Not applicable to Online Data Proxy) Configure the port to receive synchronization requests from client devices.

Prerequisites

A secure synchronization stream uses SSL or TLS encryption. Both TLS and SSL require production-ready certificates to replace the default ones installed with SAP Mobile Server. Ensure that you possess digital certificates verified and signed by third-party trusted authorities. See *Encrypting Synchronization for Replication Payloads* in *Security*.

Task

1. In the left navigation pane, select **Configuration**.
2. In the right administration pane, click the **General** tab.
3. From the menu bar, select **Components**.
4. Select **Replication** and click **Properties**.

5. Select the protocols and ports you require. You can enable any combination of HTTP, HTTPS and mutual HTTPS, but you must enable at least one port.
 - (No encryption): if you do not require SSL encryption, choose **Port**. SAP Mobile Platform recommends this option if you do not require a secure communication stream for synchronization. By default, the port for HTTP is 2480.
 - (Server authentication): to encrypt the HTTP stream with SSL for authentication, choose **Secure port**. By default, the port for authentication over HTTPS is 2481. The "Secure Sync Port" properties can be used to review and set the server identity and public certificate for the secure synchronization port. See below.
 - (Mutual authentication): to encrypt the HTTP stream with SSL for transport, choose **Mutual secure port**. By default, the port for transport over HTTPS is 2482.
6. (Optional) Configure additional properties for E2EE with TLS, HTTPS with SSL, and synchronization server startup options:

Note: Leave E2E Encryption values blank to disable end-to-end encryption.

- E2E Encryption Type – specify the asymmetric cipher used for key exchange for end-to-end encryption. You can only use RSA encryption.
- E2E Encryption Certificate Alias – specify E2E encryption key pair files alias in the key store. This value must be an alias configured to have a private key.
- Secure Sync Port Certificate Alias – specify the HTTPS identity file alias in the key store. The value must be an alias configured to have a private key.
- Trusted Relay Server Certificate Alias – specify the relay server trusted certificate alias in the keystore. The value can be any alias in the server key store.
- User Options – sets the command line options for starting the synchronization server. These options are appended the next time the synchronization server starts. For a complete list of user options, see <http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.help.sqlanywhere.12.0.1/mlserver/mobilink-ml-syncserver.html>
Do not use the User Options property in SAP Control Center to pass in these options: -c, -lsc, -q, -w, -x, -zs.

7. Click **OK**.

When you set the **E2E Encryption Certificate Alias** or the **Secure Sync Port Certificate Alias**, a public key and private key are automatically generated from the selected key store alias.

Configuring Secure Communication With an HTTP Proxy

If you want SAP Mobile Server connect to an HTTP Proxy, you can set connection properties for it when you optimize SAP Mobile Server performance.

1. In the left navigation pane, select **Configuration**.
2. In the right administration pane, select the **General** tab.
3. From the menu bar, select **Performance**.

4. Select **Java Virtual Machine**, then click **Properties**.
5. In the User Options row, enter command-line options to control the startup behavior. Enter options as a series of Java system property and value pairs, using this syntax:

```
-DpropertyName=value
```

Supported properties include:

- **-Dhttp.proxyHost** – for the host name of the proxy server.
- **-Dhttp.proxyPort** – for the port number. The default value is 80.
- **-Dhttp.nonProxyHosts** – for the list of hosts that should be reached directly, thereby bypassing the proxy. Separate multiple entries with |.

The patterns may start or end with a * for wildcards. A host name that matches the wildcard pattern can bypass the proxy. For example, to use command line options to configure a proxy and other non-proxy hosts (including those on local computers):

```
-Dhttp.proxyHost=proxy.myDomain.com -Dhttp.proxyPort=8080 -
Dhttp.nonProxyHosts=*.myOtherDomain1.com|localhost|
*.myOtherDomain2.corp
```

Note: You can throttle the number of connections used by SAP Mobile Platform by setting a proxy connection pool size on each SAP Mobile Server node. You can set the HTTP proxy configuration in the user options of the Java Virtual Machine properties. The maximum number of gateway connections is determined by this equation:

$$number_nodes * connection_pool_size$$

For details, see *Proxy Properties*.

6. Click **Save**.
7. Restart the server for changes to take effect.

See also

- *Proxy Properties* on page 192

Messaging

Messaging is a synchronization method used to maintain data integrity on device client applications. It uses a JMS service to upload and download data changes to and from the SAP Mobile Server cache database. Messaging-based synchronization ports implement a strongly encrypted HTTP-based protocol using a proprietary method.

Configure messaging in the Messaging tab of the Server Configuration node for the particular server you are administering.

See also

- *Configuring Management Port Properties* on page 42
- *Replication* on page 43
- *Enforcing OData Application Registration* on page 46

- *Upgrade: Configuring Data Change Notification Components* on page 47
- *Configuring Cluster Performance Properties* on page 47
- *Configuring SSL Properties* on page 51
- *Configuring Web Container Properties* on page 56
- *Configuring Cache Properties* on page 58
- *Configuring Server Log Settings* on page 59

Configuring Messaging Properties

Configure a port to receive service requests from devices.

1. In the left navigation pane, select **Configuration**.
2. In the right administration pane, click the **General** tab.
3. From the menu bar, select **Components**.
4. Select **Messaging**, then click **Properties**.
5. Enter the synchronization port number. The default is 5001.
6. (Optional) Configure the BES response portioning value.

There are limits on the amount of data that can be downloaded through an HTTP connection using the BlackBerry MDS. The BES response portioning limit determines the amount of HTTP traffic the BES MDS server accepts from SAP Mobile Server. For BlackBerry MDS, this limit is set in BlackBerry Manager using the Maximum KB/Connection setting.

7. Click **OK**.

Enforcing OData Application Registration

The Administrator can require OData applications to register with the SAP Mobile Server cluster.

1. In the left navigation pane, select **Configuration**.
2. In the right administration pane, click the **General** tab.
3. From the menu bar, select **Components**.
4. Select **OData Client**, then click **Properties**.
5. To require all OData applications to register, select the **All proxy applications need to register** checkbox.
6. click **OK** and **Save** on the Components tab.
7. Restart SAP Mobile Server for changes to take effect.

See also

- *Configuring Management Port Properties* on page 42
- *Replication* on page 43
- *Messaging* on page 45

- *Upgrade: Configuring Data Change Notification Components* on page 47
- *Configuring Cluster Performance Properties* on page 47
- *Configuring SSL Properties* on page 51
- *Configuring Web Container Properties* on page 56
- *Configuring Cache Properties* on page 58
- *Configuring Server Log Settings* on page 59

Upgrade: Configuring Data Change Notification Components

Configure the behavior of Data Change Notification (DCN) components in the cluster. You can configure DCN to accept HTTP GET and POST requests to support applications developed with versions previous to SAP Mobile Platform 2.2. By default, the current version of SAP Mobile Server only accepts HTTP POST requests for DCN.

1. In the left navigation pane, select **Configuration**.
2. In the right administration pane, click the **General** tab.
3. From the menu bar, select **Components**.
4. If you want SAP Mobile Server to be backward compatible with DCN code written previous to version 2.2., select **Data Change Notification** and click **Properties**.
5. Select the **Enable HTTP GET method** text box.
By setting this property, developers do not need to update the code of existing applications.

See also

- *Configuring Management Port Properties* on page 42
- *Replication* on page 43
- *Messaging* on page 45
- *Enforcing OData Application Registration* on page 46
- *Configuring Cluster Performance Properties* on page 47
- *Configuring SSL Properties* on page 51
- *Configuring Web Container Properties* on page 56
- *Configuring Cache Properties* on page 58
- *Configuring Server Log Settings* on page 59

Configuring Cluster Performance Properties

To optimize SAP Mobile Server performance across the cluster, configure performance properties.

See also

- *Configuring Management Port Properties* on page 42
- *Replication* on page 43
- *Messaging* on page 45

- *Enforcing OData Application Registration* on page 46
- *Upgrade: Configuring Data Change Notification Components* on page 47
- *Configuring SSL Properties* on page 51
- *Configuring Web Container Properties* on page 56
- *Configuring Cache Properties* on page 58
- *Configuring Server Log Settings* on page 59

Configuring Replication Performance Properties

Configure performance properties for replication synchronization in the cluster.

1. In the left navigation pane, select **Configuration**.
2. In the right administration pane, click the **General** tab.
3. From the menu bar, select **Performance**.
4. Select **Replication** and click **Properties**.
5. Configure the following properties:
 - Synchronization cache size – maximum size for the replication protocol server memory cache.
 - Synchronization thread count – MobiLink thread count. This value should be lower than the thread count for the SQL Anywhere database.
6. Click **OK**.

Configuring Messaging Performance Properties

Configure performance properties for messaging-based synchronization in the cluster.

1. In the left navigation pane, select **Configuration**.
2. In the right administration pane, click the **General** tab.
3. From the menu bar, select **Performance**.
4. Select **Messaging** and click **Properties**.
5. Configure the following properties:
 - Maximum number of in memory messages – maximum allowable number of in memory messages.
 - Subscribe bulk load thread pool size – maximum number of threads allocated to initial bulk load subscription operations. The default value is 5. Setting the thread pool size too high can impact performance.
 - Inbound messaging queue count – number of message queues used for incoming messages from the messaging-based synchronization application to the server. SAP recommends you choose a value that represents at least 10% of active devices.
 - Outbound messaging queue count – number of message queues used for outbound messages from the server to the messaging-based synchronization application. SAP

recommends a value that represents at least 50% of active devices. However, if you are running 32-bit operating system, do not exceed a value of 100% of active devices.

- Asynchronous operation replay queue count – the queue limit for asynchronous operation replays. The minimum acceptable queue count is 1 and the default is 5. .

6. Click **OK**.

Configuring Asynchronous Operation Replay Queue Count

Configure properties of a cluster to control whether asynchronous operation replays are enabled for all cluster packages.

1. In the left navigation pane, select **Configuration**.
2. In the right administration pane, click the **General** tab.
3. From the menu bar, select **Performance**.
4. Select **Messaging** and click **Properties**.
5. Configure the queue limit for asynchronous operation replays in **Asynchronous operation replay queue count**. The minimum acceptable queue count is 1 and the default is 5.
6. Click **Save**.

See also

- *Viewing Asynchronous Operation Replays* on page 356

Configuring Connection Performance Properties

Configure performance properties for HTTP and synchronous message services connections in the cluster.

1. In the left navigation pane, select **Configuration**.
2. In the right administration pane, click the **General** tab.
3. From the menu bar, select **Performance**.
4. Select **Connection Global** and click **Properties**.
5. Configure the following properties:
 - Maximum count of webservice connections per host – maximum number of web service connections allowed per host configuration.
 - Webservice connection timeout(s) – length of time, in seconds, until a web service connection is established. A value of 0 (zero) means no timeout.
 - Maximum proxy connection pool size – maximum size for the replication protocol server memory cache.
 - Proxy connection timeout(s) – length of time, in milliseconds, until a proxy connection is established. A value of 0 (zero) means no timeout.
 - Maximum count of total webservice connections – maximum number of web service connections allowed overall.

6. Click **OK**.

Configuring Java Virtual Machine Performance Properties

To optimize Java Virtual Machine performance for all servers in the cluster, configure the maximum and minimum heap sizes, thread stack size, and user options such as enabling trace upload to SAP® Solution Manager.

1. In the left navigation pane, select the **Configuration** tab.
2. In the right administration pane, click the **General** tab.
3. From the menu bar, select **Performance**.
4. Select **Java Virtual Machine**, then click **Properties**.
5. Configure these Java Virtual Machine performance properties, as required:
 - **Maximum Heap Size** – the maximum size of the JVM memory allocation pool. Use K to indicate kilobytes, M to indicate megabytes, or G to indicate gigabytes. For production recommendations on this value, see *SAP Mobile Server Replication Tuning Reference* in *System Administration*.
 - **Minimum Heap Size** – the minimum size of the JVM memory allocation pool, in megabytes. For production recommendations on this value, see *SAP Mobile Server Replication Tuning Reference* in *System Administration*.
 - **Thread Stack Size** – the JVM `-Xss` option.
 - **User Options** – other JVM options.

For example, you can enable JVM garbage collection logging by setting `-XX: +PrintGCDetails`.

If you are in an SAP environment and want to define the Solution Manager URL into which trace files can be uploaded, use this user option: `-`

```
Dcom.sap.solutionmanager.url=<SolutionManager_URL>.
```

Note: Enter options as a series of Java system property and value pairs, delimited by spaces, with this syntax:

```
-DpropertyName1=value1 -DpropertyName2=value2
```

For information on troubleshooting JVM settings, see *SAP Mobile Server Fails to Start Due to Incorrect JVM Settings* in *Troubleshooting*.

To update the disk cache size, use the `-ds` user option. The disk cache size must be followed by k, m, or g, for example, `-ds 10G`.

6. Click **Ok**.

Configuring Client Dispatcher Performance Properties

Configure the dispatcher for SAP Mobile Platform client traffic.

1. In the left navigation pane, select **Configuration**.

2. In the right administration pane, click the **General** tab.
3. From the menu bar, select **Performance**.
4. Select **Client Dispatcher** and click **Properties**.
5. Configure the thread count for the messaging client dispatcher.
6. Configure the following additional properties for the messaging client dispatcher:
 - Max connections – maximum number of messaging client connections allowed per host configuration.
 - Connection timeouts – length of time, in seconds, after which a client connection request times out.
 - Connection idle times – length of time, in seconds, that a client connection can remain idle before the socket is closed.
 - Response timeouts – length of time, in seconds, for a response to a client connection request to be received, after which the connection request times out.
 - Thread count – the thread count in the thread pool for messaging client connections.
7. Click **OK**.

Configuring SSL Properties

Configure SSL certificates and security profiles to facilitate Secure Sockets Layer (SSL) encryption for communication ports in SAP Mobile Server.

Prerequisites

Ensure you have set up the server environment before you configure a security profile as part of the server configuration. For more information, see *Encrypting Synchronization with SSL for Replication* in the *Security* guide.

1. *Managing Certificates for SSL Encryption*

Manage keystore certificates to be used for SSL encryption of SAP Mobile Server communication ports. All security profiles use the same keystore.

2. *Managing Ciphers for SSL Encryption*

Define the ciphers supported in the security characteristic/security profile. You may not want to support all ciphers, for example, if not required or not considered secure.

3. *Creating an SSL Security Profile in SAP Control Center*

Security profiles define the security characteristics of a client/server session. Assign a security profile to a listener, which is configured as a port that accepts client connection requests of various protocols. SAP Mobile Server uses multiple listeners. Clients that support the same characteristics can communicate to SAP Mobile Server via the same port defined in the listener.

4. *Enabling OCSP*

(Optional) Enable OCSP (Online Certificate Status Protocol) to determine the status of a certificate used to authenticate a subject: current, expired, or unknown. OCSP configuration is enabled as part of cluster level SSL configuration. OCSP checking must

be enabled if you are using the `CertificateAuthenticationLoginModule` and have set `Enable revocation checking` to `true`.

See also

- *Configuring Management Port Properties* on page 42
- *Replication* on page 43
- *Messaging* on page 45
- *Enforcing OData Application Registration* on page 46
- *Upgrade: Configuring Data Change Notification Components* on page 47
- *Configuring Cluster Performance Properties* on page 47
- *Configuring Web Container Properties* on page 56
- *Configuring Cache Properties* on page 58
- *Configuring Server Log Settings* on page 59

Managing Certificates for SSL Encryption

Manage keystore certificates to be used for SSL encryption of SAP Mobile Server communication ports. All security profiles use the same keystore.

1. In the left navigation pane, select **Configuration**
2. In the right administration pane, select the **General** tab.
3. From the menu bar, select **SSL Configuration**.
4. To manage the keystore configuration for SSL encryption for all security profiles, select **Key Store Configuration**.
You see a list of all certificates in the server keystore, and can import, export, or delete certificates.

Next

Create an SSL security profile that uses the selected certificates.

See also

- *Managing Ciphers for SSL Encryption* on page 54

Importing Certificates

Import a certificate for SSL encryption from a local file system.

You can import two types of certificates into the server key store.

- X.509 format supports storage of a single certificate. This format does not support storage of the private key or certificate or certification path. No private key password is required.
- Personal Information Exchange format (PFX, also called PKCS #12) supports secure storage of certificates, private keys, and all certificates in a certification path. A private key password is required.

1. Click **Import**.
2. Select the Certificate type as X.509 or PKCS #12.
3. Enter a unique **Alias** name for the certificate.
Each alias identifies a certificate in the server key store. You can assign an alias to a security component for reference.
4. Click **Browse** and select an X.509 or PKCS #12 certificate to import into the key store.
 - For an X.509 certificate, the expected file extension is .cer or .crt.
 - For a PKCS #12 certificate, the expected file extension is .pfx or .p12.
5. For PKCS #12 certificates, enter the **Private key password** for the private key to be loaded from a .pfx or .p12 file.

Exporting Certificates

Export a certificate for SSL encryption to a local file system.

You can export two types of certificates, X.509 and PKCS #12.

- For a key entry which has a private key, you can export the entry as an X.509 or PKCS #12 certificate.
- For a key entry which has no private key, you can only export the entry as an X.509 certificate.

1. Select a key store entry from the Key Store Properties dialog.
2. Click **Export**.
3. If the certificate has a private key:
 - a) Select the **Certificate type**.
 - b) Enter and confirm the password to protect the private key in the exported .p12 file.
4. Click **Next**.
5. Click **Finish** to save the certificate to a file.
6. Select a location to export the file to the local file system.

An exported X.509 certificate is saved as a `certificate_<alias>.cer` file.

An exported PKCS #12 certificate is saved as a `certificate_<alias>.p12` file.

Deleting Certificates

Delete a certificate for SSL encryption from the server key store.

You cannot delete a certificate if the private key is in use.

1. Select a certificate in the Key Store Properties dialog.
2. Click **Delete**.
3. Confirm that you want to delete the selected certificate.

Managing Ciphers for SSL Encryption

Define the ciphers supported in the security characteristic/security profile. You may not want to support all ciphers, for example, if not required or not considered secure.

1. In the left navigation pane, select **Configuration**
2. In the right administration pane, select the **General** tab.
3. From the menu bar, select **SSL Configuration**.
4. To manage the supported ciphers for SSL encryption for all security profiles, select **Ciphers**.
You see a list of all available and enabled ciphers.
5. Use the **Add**, **Add all**, **Remove**, and **Remove all** buttons to add or remove ciphers from the enabled list.
Only Enabled Ciphers are supported for the SSL Configuration.

Note: Either one or both of `SSL_RSA_WITH_3DES_EDE_CBC_SHA` and `TLS_RSA_WITH_AES_128_CBC_SHA` must be enabled since these are essential ciphers.

6. Click **OK** to save your changes.
7. Restart SAP Mobile Server for your changes to take effect.

See also

- *Managing Certificates for SSL Encryption* on page 52

Creating an SSL Security Profile in SAP Control Center

Security profiles define the security characteristics of a client/server session. Assign a security profile to a listener, which is configured as a port that accepts client connection requests of various protocols. SAP Mobile Server uses multiple listeners. Clients that support the same characteristics can communicate to SAP Mobile Server via the same port defined in the listener.

Note: A security profile can be used by one or more servers in a cluster, but cannot be used by multiple clusters.

1. In the left navigation pane, select **Configuration**
2. In the right administration pane, select the **General** tab.
3. From the menu bar, select **SSL Configuration**.
4. In the **Configure security profile table**:
 - a) Enter a name for the security profile.
 - b) Select a certificate alias from the drop down list.

This is the alias of a key entry in the keystore. Make sure the key password of this key entry is the same as the keystore password.

c) Select a mutual SSL setting:

- **true** – the value of the securityCharacteristic is 'default_mutual'. The security profile authenticates both the client and the server, the client is also required to authenticate using a certificate; both the client and server will provide a digital certificate to be accepted or rejected by the other.
- **false** – the value of securityCharacteristic is 'default'. The security profile authenticates only the server, only the server must provide a certificate to be accepted or rejected by the client.

5. Click **Save**.

6. From the **Components** menu, assign the security profile to the desired management or communication ports.

Enabling OCSP

(Optional) Enable OCSP (Online Certificate Status Protocol) to determine the status of a certificate used to authenticate a subject: current, expired, or unknown. OCSP configuration is enabled as part of cluster level SSL configuration. OCSP checking must be enabled if you are using the CertificateAuthenticationLoginModule and have set Enable revocation checking to true.

Enable OCSP for a cluster when configuring SSL.

1. In the left navigation pane, select **Configuration**.
2. In the right administration pane, select the **General** tab.
3. From the menu bar, select **SSL Configuration**.
4. To enable OCSP when doing certificate revocation checking, check **Enable OCSP**.
5. Configure the responder properties (location and certificate information):

Responder Property	Details
URL	A URL to responder, including its port. For example, <code>https://ocsp.example.net:80</code> .

Responder Property	Details
Certificate subject name	The subject name of the responder's certificate. By default, the certificate of the OCSP responder is that of the issuer of the certificate being validated. Its value is a string distinguished name (defined in RFC 2253), which identifies a certificate in the set of certificates supplied during cert path validation. If the subject name alone is not sufficient to uniquely identify the certificate, the subject value and serial number properties must be used instead. When the certificate subject name is set, the certificate issuer name and certificate serial number are ignored. For example, CN=MyEnterprise, O=XYZCorp.
Certificate issuer name	The issuer name of the responder certificate. For example, CN=OCSP Responder, O=XYZCorp.
Certificate serial number	The serial number of the responder certificate.

Configuring Web Container Properties

Configure and tune HTTP ports and some general properties.

1. In the left navigation pane, select **Configuration**.
2. In the right administration pane, click the **Web Container** tab.
3. Configure the following web server properties, as required:
 - Max form content size – amount of data that can post back from a browser or other client to the server. The default is 10000000 bytes, with a maximum of 100000000 bytes.
 - Perform server side compression – enables HTTP compression in a production environment when client demand becomes high and affects server performance for ODP applications.

Note: If you enable server compression, notify developers that client behavior must accommodate compression: namely, the client needs to use the Encoding parameter for HTTP headers and set the value to GZIP Information is available on the Internet if

more information is required. See *ODPRequest class* in *Developer Guide: OData SDK*.

4. To create a new HTTP/HTTPS port, click **New**.
5. Configure the port properties, as required:
 - Port – port number where the server listens for requests.
 - Status – whether the port is enabled.
 - Protocol – communication protocol. When **Protocol** is set to https, the **Security Profile** drop down is displayed to allow selection of a security profile.
 - Maximum threads – maximum number of active threads that the server allows for this listener. This value must be less than or equal to the allowed number of connections.
 - Statistics on – enables statistics collection on connections.
 - Maximum idle time – the length of time, in seconds, that a socket is idle before it disconnects.
 - So linger time(s) – the length of time, in seconds, that a socket lingers before it disconnects to enable a graceful shutdown.
 - Acceptors number – number of acceptor threads to be run.
Do not set this number too high. Use the following method to determine the recommended number: `Math.max(1, (<processesCount or coresCount>+3) / 4)`.
 - Acceptor priority – priority of the acceptor threads relative to the other threads. The priority is adjusted by the value you enter here (the default 0) to either favor the acceptance of new threads and newly active connections, or to favor the handling of already dispatched connections.
 - Response buffer size (k) – size of the content buffer for sending responses. These buffers are used only for active connections that are sending responses with bodies that do not fit within the header buffer.
 - Request buffer size (k) – size of the content buffer for receiving requests. These buffers are used only for active connections that have requests with bodies that do not fit within the header buffer.
 - Header buffer size (k) – size of the buffer to be used for request and response headers. An idle connection has a maximum of one buffer of this size allocated.
 - Listen backlog – maximum backlog for incoming connections.
 - Low resources connections – number of connections, which if exceeded, places this connector in a low resources state. This is not an exact measure, as the connection count is averaged over the select sets. When in a low resources state, different idle timeouts can apply on connections as specified in the low resources maximum idle time property.
 - Low resources maximum idle time(s) – period, in milliseconds, that a connection is allowed to be idle when there are more connections than the number set for the low resources connections property. This allows the server to rapidly close idle connections to gracefully handle high load situations.

- Include server info in session – includes the server's IP address and port in the HTTP session ID. This is required when using the Web redirectors for load balancing to ensure sticky sessions.

6. Click **OK**.

7. Click **Save**.

See also

- *Configuring Management Port Properties* on page 42
- *Replication* on page 43
- *Messaging* on page 45
- *Enforcing OData Application Registration* on page 46
- *Upgrade: Configuring Data Change Notification Components* on page 47
- *Configuring Cluster Performance Properties* on page 47
- *Configuring SSL Properties* on page 51
- *Configuring Cache Properties* on page 58
- *Configuring Server Log Settings* on page 59

Configuring Cache Properties

A configuration cache acts as the distributed memory used by all nodes that are member of a cluster. Configure general properties for this cache.

1. In the left navigation pane, select **Configuration**.
2. In the right administration pane, click the **Configuration Cache** tab.
3. From the menu bar, select **Cache**.
4. Configure these properties, as required:
 - Auto increase port number – set to `true` to increment the cache port number.
 - Encrypt configuration cache network – set to `true` to encrypt data at the transport layer.
 - Port – port number for the configuration cache.
5. Click **Save**.
6. From the menu bar, select **Push cache**.
7. Configure these properties, as required:
 - Auto increase port number – set to `true` to increment the cache port number.
 - Core pool size – core number of threads for the executor service.
 - Encrypt configuration cache network – set to `true` to encrypt data at the transport layer.
 - Keep alive time(s) – length of time, in seconds, that threads can remain idle before being terminated.
 - Maximum pool size – maximum allowed number of threads for the executor service.
 - Port – port number for the configuration cache.

8. Click **Save**.

See also

- *Configuring Management Port Properties* on page 42
- *Replication* on page 43
- *Messaging* on page 45
- *Enforcing OData Application Registration* on page 46
- *Upgrade: Configuring Data Change Notification Components* on page 47
- *Configuring Cluster Performance Properties* on page 47
- *Configuring SSL Properties* on page 51
- *Configuring Web Container Properties* on page 56
- *Configuring Server Log Settings* on page 59

Configuring Server Log Settings

Server logs collect data that enables you to monitor system health. Configure the server log settings to specify the amount of detail that is written to the log.

See also

- *Configuring Management Port Properties* on page 42
- *Replication* on page 43
- *Messaging* on page 45
- *Enforcing OData Application Registration* on page 46
- *Upgrade: Configuring Data Change Notification Components* on page 47
- *Configuring Cluster Performance Properties* on page 47
- *Configuring SSL Properties* on page 51
- *Configuring Web Container Properties* on page 56
- *Configuring Cache Properties* on page 58

Configuring SAP Mobile Server Log Settings

SAP Mobile Server logs collect data on SAP Mobile Server health and performance by component. Configure SAP Mobile Server log properties to specify the amount of detail that is written to the log, as well as the duration of the server log life cycle.

Additionally, you should always use SAP Control Center to configure server logs. If you manually edit the configuration file, especially on secondary servers in a cluster, the servers may not restart correctly once shut down.

1. In the SAP Control Center left navigation pane, click **Configuration**.
2. In the right administration pane, click the **Log Setting** tab and select **Mobile Server**.

3. The option "Start a new server log on server restart" is set by default. When selected, this option means a new version of the log file is created after server restart, and the old one is archived.
4. Set the MMS server log size and backup behavior that jointly determine the server log life cycle.
 - a) Set the **Maximum file size**, in kilobytes, megabytes, or gigabytes, to specify the maximum size that a file can reach before a new one is created. The default is 10MB. Alternatively, select **No limit** to log all events in the same file, with no maximum size.
 - b) Set the **Maximum backup index** to determine how many log files are backed up before the oldest file is deleted. The index number you choose must be a positive integer between 1 and 65535. The default is 10 files. Alternatively, select **No limit** to retain all log files.
5. Set the HTTP log settings.
 - a) Select **Enable HTTP request log** to generate an HTTP request log in the logs subdirectory. The generated log file name is `server-name-http.log`.

Note: HTTP logging is off by default. Enabling HTTP logging can cause a performance impact and possible logging of sensitive data.

 - b) Set the **Maximum file size** of the log file.

Note: Maximum file size only takes effect when you select either **Perform rotation** or **Archive**.

 - c) If you want to back up the log file when it reaches the limit, select **Perform rotation**. The backup file is saved as `server-name-http.log.backup-number`.
 - d) If you want to continue to use the current log file when the server restarts, select **Reuse**. If you do not select this option, when the server restarts the current log file is copied to the `.\old` subdirectory and a new log file is created.
 - e) If you want to archive the log file select **Archive**, then specify the **Archive file name**. If you want to compress the archive log file select **Compress**.
6. For each component, choose a log level:

Component	Default Log Level
MMS	Info
PROXY	Info
Cluster	Info
MSG	Info
Security	Info
PUSH	Info

Component	Default Log Level
Mobilink	Info
DataServices	Info
Other	Warn
DOEC	Info

Log level	Messages logged
All	Complete system information
Trace	Finer-grained informational events than debug
Debug	Very fine-grained system information, warnings, and all errors
Info	General system information, warnings, and all errors
Warn	Warnings and all errors
Error	Errors only
Console	Messages that appear in the administration console only (when SAP Mobile Server is running in non-service mode)
Off	Do not record any messages

7. Click **Save**.

Log messages are recorded as specified by the settings you choose. The log file is located in: `SMP_HOME \Servers\UnwiredServer\logs\<hostname>-server.log`.

Log life cycle default example

If you keep the default maximum file size and default index, an SAP Mobile Server writes to the log file until 10MB of data has been recorded. As soon as the file exceeds this value, a new version of the log file is created (for example, the first one is `<hostname>-server.log.1`). The contents of the original log are backed up into this new file. When the `<hostname>-server.log` file again reaches its limit:

1. The contents of `<hostname>-server.log.1` are copied to `<hostname>-server.log.2`.
2. The contents of `<hostname>-server.log` are copied to `<hostname>-server.log.1`.
3. A new copy of `<hostname>-server.log` is created.

This rollover pattern continues until the backup index value is reached, with the oldest log being deleted. If the backup index is 10, then `<hostname>-server.log.10` is the file removed, and all other logs roll up to create room for the new file.

Configuring Messaging Server Log Settings

Messaging Server logs create trace configurations for messaging modules, and retrieve trace data for all or specific messages. Configure trace configuration properties for modules to specify the amount of detail that is written to the log. You can configure trace settings for the primary server cluster in SAP Control Center for each module. The settings are available to cluster servers through the shared data folder.

Note: The default settings may only need to change in case of technical support situations where, for diagnostic reasons, a request is made to configure the specific module(s) settings, and provide the request log. In all other cases, the administrator or developer should not need to change the settings.

Additionally, you should always use SAP Control Center to configure server logs. If you manually edit the configuration file, especially on secondary servers in a cluster, the servers may not restart correctly once shut down.

1. In the SAP Control Center left navigation pane, click **Configuration**.
2. In the right administration pane, click the **Log Setting** tab and select **Messaging Server**.
3. Select Default, or one or more of the messaging service modules. Click **Show All** to show all modules.

Module	Description
Default	Represents the default configuration. The default values are used if optional fields are left blank in a module trace configuration. Required.
Device Management	Miscellaneous functions related to device registration, event notification, and device administration. Enable tracing for problems in these areas.
JMSBridge	This module handles communications from the SAP Mobile Server to the messaging server. Enable tracing to view the detailed messaging exchange.
MO	This module handles the delivery of messages between the client and server, including synchronous function calls from client to server. Enable tracing for MO errors and message delivery issues.

Module	Description
SUPBridge	This module handles communications from the messaging server to the SAP Mobile Server. Enable tracing to view the detailed messaging exchange.
TM	This module handles the wire protocol, including encryption, compression, and authentication, between the messaging server and clients. All communication between the client and the messaging server passes through TM. Enable tracing for authentication issues, TM errors, and general connectivity issues.
WorkflowClient	The WorkflowClient module.

4. Click **Properties**.

- a) Enter trace configuration properties. If you selected multiple modules, a string of asterisks is used to indicate settings differ for the selected modules. You can select the option to view or change the property value for any module.

Property	Description
Module	Display only. Default, module name, or list of module names selected.
Description	(Optional) Custom description of the server module.
Level	Trace level for the module - DISABLED, ERROR, WARN, INFO, DEBUG, DEFAULT. If the default trace level is specified for the module, the module uses the trace level defined for Default. Required.
Max trace file size	(Optional) Maximum trace file size in MB. If the trace file size grows larger than the specified value, the trace file data is backed up automatically.
User name	(Optional) Only data for the specified user name is traced.
Application Connection ID	(Optional) Only data for the specified Application ID is traced.

- b) Click **OK**.

Log files for each module are stored in folders of the same name located in:
SMP_HOME\Servers\UnwiredServer\logs.

Viewing Cluster Information

View cluster information to determine the name and size of the cluster.

1. In SAP Control Center navigation pane, click the name of the cluster.
2. Review information for general properties:
 - The name of the cluster. By default the cluster name is `mysupcluster`.
 - The number of servers that are members of the cluster.
 - The name of servers that have outbound enablers configured.
 - The asynchronous operation replay queue count. See *Configuring Asynchronous Operation Replay Queue Count*.

See also

- *Viewing Asynchronous Operation Replays* on page 356

Checking System Licensing Information

Review licensing information to monitor used mobile device licenses, license expiry dates, and other license details. This information allows administrators to manage license use and determine whether old or unused device licenses should be transferred to new devices.

1. In the left navigation pane, select the top-level tree node.
2. In the right administration pane, select the **General** tab, and click **Licensing**.
3. Review the following licensing information:
 - Server license type – the type of license currently used by SAP Mobile Platform. For more information on license types, see *SAP Mobile Platform License Types* in *Landscape Design and Integration*.
 - Production edition – the edition of the software you have installed.
 - Server license expiry date – the date and time at which the server license expires. When a server license expires, SAP Mobile Server generates a license expired error and SAP Mobile Server is stopped.
 - Overdraft mode – allows you to generate additional licenses in excess of the quantity of licenses you actually purchased. This enables you to exceed your purchased quantity of licenses in a peak usage period without impacting your operation. This mode is either enabled or disabled, as specified by the terms of the agreement presented when you obtain such a license.
 - Used mobile user license count – the number of mobile user licenses currently in use. A mobile user is a distinct user identity—username and associated security configuration—that is registered in the server. As such, the used mobile user license count represents the total distinct user identities registered on the server. One mobile user may access:
 - Multiple applications and different versions of the same application.

- The same or different versions of an application from multiple devices.

4. Click **Close**.

Note: SAP Mobile Platform licensing is configured during installation. However, if necessary, license details can be changed at a later time.

Checking Cluster Status

Verify that a cluster is running.

In the left navigation pane, check the status (in brackets) beside the cluster name.

Managing Service Pack Information

View service pack and patch information to allow you to determine what service packs and patches have been installed, and to help troubleshoot upgrade problems.

Viewing Installed Patches

View a list of installed and missed patches.

1. In the left navigation pane, select the top-level tree node.
2. In the right administration pane, select the **Product Information**.
3. From the menu bar, select **General**.
4. View the list of installed and missed patches.

In a cluster environment, information is retrieved for all nodes in this cluster, including the data tier node. The "Server Status" column shows the result of pinging the current server. As the data tier node has no server, its status is empty.

5. To see more details about a patch, select the patch and click **Review Patch Detail**. SAP Control Center displays detail for the components of the current server node.

Viewing Installation History

View a history of installed patches for a server node

1. In the left navigation pane, select the top-level tree node.
2. In the right administration pane, select the **Product Information**.
3. From the menu bar, select **Patch Installed History**.
4. Select the **Server Node** from the dropdown menu, and click **Retrieve**. SAP Control Center displays a history of installed patches for the server node.

Exporting Product Information

Export the product information to a text file.

1. In the left navigation pane, select the top-level tree node.
2. In the right administration pane, select the **Product Information**.
3. From the menu bar, select **General**.
4. Click **Export**.
5. Click **Next**.

Product information is exported into a text file. The file contains information for each server node, expressed in a node hierarchy. Each server node includes the server name, product version, and installed and missed patches.

Sharing Cluster Information With SAP Servers

If your SAP Mobile Platform deployment is part of a larger SAP® landscape, review the SLD servers with which cluster information can be shared.

To share cluster information requires one of these SAP servers. Depending on the server type, you must either register the server, or export information to it.

SLD Server Registration

System Landscape Directory (SLD) is a central repository of system landscape information used to manage the software lifecycle.

SLD describes the systems and software components that are currently installed. SLD data suppliers register the systems on the SLD server, and keep the information up-to-date. SAP Mobile Platform is a third-party system that must be registered with SLD.

To prepare the SLD server environment for SAP Mobile Platform, ensure the following prerequisites:

- The installed version of SLD is for SAP NetWeaver 7.0 (2004s) SPS07 or higher.
- The SLD server is running.
- The SLD is configured to receive data. For more information, see the *Post-Installation Guide* and the *User Manual* for your SAP NetWeaver version on SDN: <http://www.sdn.sap.com/irj/sdn/nw-sld>.
- You contact the SLD administrator and determine the connection values to the SLD server, including its host name, protocol (HTTP or HTTPS), HTTP(S) port and the SLD user account.
- The SLD to which you register SAP Mobile Platform must be the latest Common Information Model version (currently 1.6.30).

Registering or Reregistering SLD Server Destinations

Registering an SLD destination identifies the connection properties needed to deliver the payload. You can register multiple destinations as required by your SAP environment. If your SLD server properties change, you must update properties as required and reregister the server with new values.

For information about SLD, see *Configuring, Working with and Administering System Landscape Directory* on <http://www.sdn.sap.com/irj/sdn/nw-sld>.

1. In the navigation pane of SAP Control Center, select the cluster name.
2. In the administration pane, click the **System Landscape Directory** tab.
3. Click **Servers**.
4. Choose one of the following:
 - If you are creating a new destination, click **New**.
 - If you are updating an existing destination, select the destination name in the table, and click **Properties**.
5. Configure the connection properties:

User Name	User name for the SLD server.
Password and Confirm Password	The user account password used to authenticate the user name entered. Password and Confirm Password must match for the password to be accepted.
Host	The host name or the IP address of the SLD server.
Port	The HTTP(S) port on which the SLD server is running. Enter a valid port number in the range of 0-65535.
Use secure	Select if you are using HTTPS protocol.

6. To validate the configuration, click **Ping**.
 7. To accept validated configuration properties, click **OK**.
- This registers the SLD destination.

Deleting a Registered SLD Server Destination

Delete an SLD server to unregister it from SAP Mobile Platform. Deleting an SLD server removes it from SAP Control Center and you can no longer use it as a payload destination.

1. In the navigation pane of SAP Control Center, click the cluster name.
2. In the administration pane, click the **System Landscape Directory** tab.
3. Click **Servers**.
4. Select one or more servers then click **Delete**.
5. In the confirmation dialog, click **Yes**.

Uploading Payloads with SAP Control Center

Use SAP Control Center to register the SLD server, and then either upload generated payloads on-demand or with a configured (and enabled) schedule.

1. *Manually Uploading or Exporting Payloads On-Demand*

Run an SLD payload generation task manually to generate and upload a payload on demand. Alternatively, export the payload to an XML file to archive SLD payload contents or to troubleshoot the cluster.

2. *Configuring and Enabling Scheduled Payload Generation and Uploads*

Configure a schedule to automatically generate a new payload that uploads to an SLD server once cluster information is aggregated from all cluster members. For information on how cluster information is aggregated and held, see *SLD and SAP Mobile Platform Architecture in System Administration*.

3. *Enabling the SLD Debug Log*

To view the SLD payload generation and upload debug level log in the SAP Mobile Server logs, set the MMS log level to **DEBUG**.

Manually Uploading or Exporting Payloads On-Demand

Run an SLD payload generation task manually to generate and upload a payload on demand. Alternatively, export the payload to an XML file to archive SLD payload contents or to troubleshoot the cluster.

1. In the navigation pane of SAP Control Center, select the name of the cluster for which you want to immediately upload an SLD payload.
2. In the administration pane, click the **System Landscape Directory** tab.
3. From the menu bar of the **System Landscape Directory** page, click **Schedule**.
4. Click **Run Now**.
The payload generation process begins.
5. Upon completion, review the contents of the payload and choose an action:
 - To export and save the contents to a file as XML, click **Save to File** and choose your file output name and location.
 - To upload the contents, select the target SLD servers and click **Finish**.

Configuring and Enabling Scheduled Payload Generation and Uploads

Configure a schedule to automatically generate a new payload that uploads to an SLD server once cluster information is aggregated from all cluster members. For information on how cluster information is aggregated and held, see *SLD and SAP Mobile Platform Architecture in System Administration*.

1. In the navigation pane of SAP Control Center, select the name of the cluster for which you want to schedule an SLD payload upload.
2. From the menu bar of the **System Landscape Directory** page, click **Schedule**.
3. To edit an existing schedule for a selected SLD server, click **Edit**.
 - a) Configure the schedule:
 - **Schedule repeat** – select how often the schedule should run. Options are **Monthly**, **Weekly**, **Daily**, **Hourly**, and **Custom**.

- If you select **Daily** or **Hourly**, specify:
 - **Start date** – select the date and time the automated upload should begin. Use the calendar picker and 24-hour time selector.
 - **End date** – select the date and time the automated upload should end.
 - **Days of the week** – select each day the automated upload schedule should run.
- If you select **Monthly** or **Weekly**, specify:
 - **Start date** – select the date and time the automated upload should begin. Use the calendar picker and 24-hour time selector.
 - **End date** – select the date and time the automated upload should end.
- Select **Custom**, to specify the interval granularity in seconds, minutes, or hours, as well as other date and time parameters.

b) Click **OK**.

4. To enable the schedule, click **Enable**.

See also

- *Enabling the SLD Debug Log* on page 69

Disabling a Schedule

You can disable a schedule that is currently enabled. Disabling a schedule prevents the payload generation process from running so that no new data is aggregated in the cluster database, nor can any current data be uploaded.

1. In the navigation pane of SAP Control Center, click the cluster name.
2. From the menu bar of the **System Landscape Directory** page, click **Schedule**.
3. Click **Disable**.

Enabling the SLD Debug Log

To view the SLD payload generation and upload debug level log in the SAP Mobile Server logs, set the MMS log level to DEBUG.

1. In the SAP Control Center left navigation pane, click **Configuration**.
2. In the right administration pane, click the **Log Setting** tab and select **Mobile Server..**
3. For the **MMS** component, select a log level of **DEBUG**.
4. Click **Save**.

See also

- *Configuring and Enabling Scheduled Payload Generation and Uploads* on page 68

Audit Measurement for SAP License Audit

For SAP built applications, administrators can generate an XML file that contains usage audit data that is then sent manually to SAP License Audit. The generated XML file is compatible

with the License Audit infrastructure. The audit data in the file includes counts of application users for the entire cluster as well as for each SAP built application that is deployed to the SAP Mobile Platform cluster.

Use SAP Control Center for SAP Mobile Platform to generate an audit measurement XML file for export to SAP License Audit. See *Sharing Application Data with SAP License Audit*, *Generating the SAP Audit Measurement File*, and *Uploading the SAP Audit Measurement File*. Also see *SAP License Audit* in *Developer Guide: SAP Mobile Server Runtime > Management API*.

Sharing Application Data with SAP License Audit

Generate an audit measurement file that includes usage data for SAP Mobile Platform and usage data for SAP applications deployed to the server.

Generate an audit measurement file that includes license data related to application usage.

1. *Generating the SAP Audit Measurement File*

Use SAP Control Center to generate an audit measurement file.

2. *Uploading the SAP Audit Measurement File*

Upload the audit measurement file to SAP License Audit by sending the file to SAP.

Generating the SAP Audit Measurement File

Use SAP Control Center to generate an audit measurement file.

Using SAP Control Center, generate a audit measurement file that can be sent to SAP for uploading to SAP License Audit.

1. In SAP Control Center, select the SAP Mobile Platform cluster and click the **General** tab.
2. Click **SAP Auditing Export**.
3. In the Export SAP Auditing Measurement window, enter the user name and click **Next**.
4. After SAP Control Center generates the file, click **Finish**.
5. Select a save location for the file and click **Save**.

Note: For information on uploading the audit measurement file to SAP License Audit, see supporting SAP documentation at <https://websmp108.sap-ag.de/licenseauditing>. Also see *Uploading the SAP Audit Measurement File*.

Uploading the SAP Audit Measurement File

Upload the audit measurement file to SAP License Audit by sending the file to SAP.

To upload the file to SAP License Audit, send the audit measurement file to SAP using the email address included in the measurement request from SAP. Included in this SAP-provided email is a link to the documentation for the SAP measurement process. See supporting SAP documentation at <https://websmp108.sap-ag.de/licenseauditing>.

Configuring SAP Solution Manager URL

Define and maintain a URL definition associated with an SAP Solution Manager instance for each application in the landscape. This endpoint is used to upload the business transaction XML generated by the client device platforms in an end-to-end trace session.

1. In the left navigation pane, select **Configuration**.
2. In the right administration pane, click the **General** tab.
3. From the menu bar, select **Components**.
4. Select **Solution Manager** and click **Properties**.
5. In the Solution Manager Component Property dialog, enter the URL associated with the appropriate SAP Solution manager.

Relay Server

Relay Server acts as a reverse proxy for client devices communicating with the SAP Mobile Server cluster, and it provides load balancing for the SAP Mobile Server cluster.

Relay Servers are deployed on the DMZ subnet. With a corresponding Outbound Enabler (RSOE), Relay Server enables communication from the SAP Mobile Server cluster to client devices, via the Internet, without opening an inbound port on the internal firewall.

Each SAP Mobile Server instance is supported by one or more RSOEs. Each RSOE opens outbound connections to the Relay Server, to handle both inbound and outbound communication channels, on behalf of the SAP Mobile Server. Connections between the RSOE and Relay Server use HTTPS protocol.

Relay Server also provides load balancing for the SAP Mobile Server cluster by forwarding requests from client devices to SAP Mobile Servers in the cluster, by round-robin distribution. However, in most production deployment environments, multiple Relay Servers are used with a third-party load balancer, which provides complete load balancing and failover capability. In this case, Relay Servers are deployed as a farm, and you need to perform additional steps at the end of the file generation task.

You must configure SAP Mobile Server to use Relay Server, using these high-level steps:

1. Use SAP Control Center to configure an SAP Mobile Server cluster with Relay Server farms, nodes and their tokens, as needed, and with Relay Server connection information.
2. Generate the Relay Server configuration file from SAP Control Center, and use it to update the Relay Server configuration (manually transfer the generated file to the Relay Server node and use **rshost.exe** utility to update the configuration). Refer to *Adding Relay Servers or Reverse Proxies* in the *Landscape Design and Integration* guide, or visit <http://dcx.sybase.com/index.html#sa160/en/relayserver/relayserver16.html> for Relay Server installation and configuration information.
3. Set up Outbound Enablers on each SAP Mobile Server node.

Configuring SAP Mobile Server to use Relay Server

Choose a method for configuring SAP Mobile Server to use Relay Server, then generate a Relay Server configuration file. Copy the file to the Relay Server host, and distribute the same configuration file to multiple Relay Server nodes.

This task applies only to a Relay Server installed on the LAN. It does not apply to the SAP Hosted Relay Server.

Note: If you are creating a custom Relay Server configuration, go to *Creating a Custom Relay Server Configuration* on page 74.

If you are using a quick configuration, continue with *Creating a Quick Configuration* on page 72.

1. Configuring Relay Server Properties

There are two methods of configuring Relay Server properties.

2. Generating the Relay Server Outbound Enabler Configuration File

To quickly and easily replicate a common outbound enabler (RSOE) configuration to multiple hosts, generate an RSOE configuration file.

3. Generating and Modifying the Relay Server Configuration File

Generate all or part of a Relay Server configuration file. Then transfer the generated file to all Relay Server hosts.

4. Setting Up RSOE

Set up one or more RSOEs for each SAP Mobile Server identified in a Relay Server configuration. The configured values are saved in the cluster database.

Configuring Relay Server Properties

There are two methods of configuring Relay Server properties.

Choose from one of these methods. Once completed, transfer the resulting configuration file to all hosts upon which Relay Server has been installed. For installation details, see *Installing a Relay Server* in *Landscape Design and Integration*.

See also

- *Generating the Relay Server Outbound Enabler Configuration File* on page 77

Creating a Quick Configuration

Create a Relay Server configuration primarily with system defaults, and create outbound enabler (OE) processes for each SAP Mobile Server.

1. In the navigation pane, click the SAP Mobile Server cluster name.
2. In the administration pane, click the **Relay Servers** tab.

3. Click **Quick Configure**.

4. Specify these property values:

Values vary for load balanced environments. If you do not configure load balancer values, outbound enablers bypass the load balancer and high availability is compromised if a direct Relay Server connection fails.

- **Host** – if the Relay Server farm has a load balancer in front of it, the host name or IP address of the load balancer. Otherwise, the host name or IP address of a single Relay Server.
- **HTTP port** – if the Relay Server farm has a load balancer in front of it, the port of the load balancer. Otherwise, the Relay Server HTTP port.
- **HTTPS port** – if the Relay Server farm has a load balancer in front of it, the port of the load balancer. Otherwise, the Relay Server HTTPS port.
- **URL suffix** – the URL suffix used by the Outbound Enabler to connect to a Relay Server. The value you set depends on whether Relay Server is installed on IIS or Apache hosts. For IIS, use `/ias_relay_server/server/rs_server.dll`. For Apache use `/srv/iarelayserver/`.
- **Replication or Messaging or Web Service (for scale-out nodes) farm token** – the security token used by the Outbound Enabler to authenticate its connection with the Relay Server. Assign a token string (up to 2048 characters); one token can be shared by all farm types. The replication, messaging, and webservice farm token values can be the same.
- **(Optional) Description** – a user-defined description of the Relay Server.

5. (Optional) Select **Advanced settings** and specify these property values:

- **HTTP user** – user name for OE authentication on the Web server (Relay Server host).
- **HTTP password** – password for OE authentication on the Web server.

6. (Optional) Configure connection values to required Internet proxy servers:

- **Proxy server host** – host name of the Internet proxy server.
- **Proxy server port** – connection port on the Internet proxy server.
- **HTTP proxy user** – user name for OE authentication on the Internet proxy server.
- **HTTP proxy password** – password for OE authentication on the Internet proxy server.

7. Click **OK** to generate a Relay Server configuration, and the OE processes for each SAP Mobile Server.

Properties in the `[backend_farm]` and `[backend_server]` sections are populated automatically, based on the SAP Mobile Server cluster name and host name.

An outbound enabler instance is automatically created for the RBS and MBS synchronization ports, and for each enabled HTTP/HTTPS port for the SAP Mobile Server host. For scale-out nodes, an outbound enabler instance is automatically created for each enabled HTTP/HTTPS port. The outbound enablers are not started.

Next

Review the values in the Relay Server configuration file, and edit if necessary.

Continue with *Generating and Modifying the Relay Server Configuration File*.

Creating a Custom Relay Server Configuration

Create a Relay Server configuration by specifying all configuration property values.

1. *Launching the Relay Server Configuration Wizard*

Launch the Relay Server Configuration wizard to create a configuration file with customized property values.

2. *Setting Relay Server General Properties*

Set basic connection properties for the Relay Server.

3. *Defining Server Farms and Cluster Nodes*

Set connection properties for the SAP Mobile Server cluster and its constituent nodes.

4. *Reviewing Configured Relay Server Properties*

Confirm the Relay Server property values before you generate the configuration file.

Launching the Relay Server Configuration Wizard

Launch the Relay Server Configuration wizard to create a configuration file with customized property values.

1. In the navigation pane, click the SAP Mobile Server cluster name.
2. In the administration pane, click the **Relay Servers** tab.
3. Click **New**.

Setting Relay Server General Properties

Set basic connection properties for the Relay Server.

Prerequisites

Launch the Relay Server configuration wizard.

Task

1. Select and specify property values for either:
 - **Standalone relay server** – outbound enablers bypass the load balancer and high availability is compromised if a direct Relay Server connection fails.
 - **Host** – the host name or IP address of a single Relay Server
 - **HTTP port** – the Relay Server HTTP port
 - **HTTPS port** – the Relay Server HTTPS port

Note: If Relay Server uses HTTPS and certificates, clients other than those using replication-based synchronization may not be able to connect: messaging

applications support only HTTP, and Hybrid Web Container applications for iOS support HTTPS, but not certificates.

- **URL suffix** – the URL suffix used by the Outbound Enabler to connect to a Relay Server. The value you set depends on whether Relay Server is installed on IIS or Apache hosts. For IIS, use `/ias_relay_server/server/rs_server.dll`. For Apache use `/srv/iarelayserver/`.

For IIS, the value identifies the *relative* path for `rs_client.dll`. If your IIS directory structure is different, modify this value accordingly.

- **(Optional) Description** – a user-defined description of the Relay Server.
- **Relay server farm** – configures a farm of relay servers, including values for the load balancer that is in front of the Relay Server farm. Values vary for load balanced environments.
 - **Load balancer host** – the host name or IP address of the load balancer in front of the Relay Server farm
 - **HTTP port** – the port of the load balancer
 - **HTTPS port** – the secure port of the load balancer
 - **URL suffix** – the URL suffix used by the Outbound Enabler to connect to a Relay Server load balancer. The value you set depends on whether Relay Server load balancer is installed on IIS or Apache hosts. The default for IIS is `/ias_relay_server/server/rs_server.dll`. The default for Apache is `/srv/iarelayserver/`.

For IIS, the value identifies the *relative* path for `rs_client.dll`. If your IIS directory structure is different, modify this value accordingly.

- **(Optional) Description** – a user-defined description of the Load Balancer.
- **Relay server nodes within farm** – The nodes of relay server within a relay server farm. Add or remove relay server nodes:

To add relay server nodes, specify these property values and click +:

- **Host** – the host name or IP address of a single Relay Server
- **HTTP port** – the Relay Server HTTP port
- **HTTPS port** – the Relay Server HTTPS port

To remove relay server nodes, select the corresponding node, then click X.

2. Add or remove HTTP credentials as required:

- a) Select **Configure relay server HTTP credentials**.
- b) To add new credentials, specify these property values and click +:
 - **User name** – user name for RSOE authentication on the Web server (Relay Server host).
 - **Password** – password for RSOE authentication on the Web server.

- c) To remove credentials from the list, select the corresponding user name, then click **X**.

3. Click Next.

Defining Server Farms and Cluster Nodes

Set connection properties for the SAP Mobile Server cluster and its constituent nodes.

1. Define the SAP Mobile Server cluster.

a) Specify these property values:

- **Farm ID** – a string that identifies the SAP Mobile Server cluster for which the Relay Server manages requests. This property is case-sensitive, and must match the value in the Outbound Enabler configuration.
- **Type** – the type of request managed by the Relay Server: Replication, Messaging or Webservice protocol. When configuring Relay Server Outbound Enabler properties for a scale-out node, you can select only the Webservice farm type.
- **(Optional) Description** – user-defined description for the SAP Mobile Server cluster.

b) Click +.

c) Repeat steps 1 and 2 to add multiple SAP Mobile Server clusters.

d) To delete a configured SAP Mobile Server cluster, select it in the list, then click the **X** button.

2. Identify each SAP Mobile Server instance in the cluster.

a) Select an existing SAP Mobile Server cluster.

b) Specify these property values:

- **Node ID** – a string that identifies the SAP Mobile Server in the cluster. This property is case-sensitive, and it must match the value in the RSOE configuration.
- **Token** – the security token used by the Outbound Enabler to authenticate its connection with the Relay Server. Assign a token string (up to 2048 characters); one token can be shared by all farm types.

c) Click +.

d) Repeat steps 1 and 2 to add SAP Mobile Server cluster nodes.

e) To delete a configured SAP Mobile Server node, select it in the list and click **X**.

3. Click Next to review your settings, or click **Finish** to exit the wizard.

The relay server configuration screen displays the Relay Server as Type STANDALONE for a standalone Relay Server or LOAD_BALANCER for a Relay Server farm.

Note: After you exit the wizard, generate the Relay Server configuration file, and copy it to each Relay Server instance to update configuration for multiple Relay Servers.

The Relay Server is registered with SAP Control Center, and can be managed from the Relay Servers tab for the SAP Mobile Server cluster.

Reviewing Configured Relay Server Properties

Confirm the Relay Server property values before you generate the configuration file.

1. Review property values to ensure that:
 - No errors exist.
 - All SAP Mobile Server clusters are defined, and assigned the correct type.
2. Click **Finish**.

The Relay Server is registered with SAP Control Center, and it can be managed from the Relay Servers tab for the SAP Mobile Server cluster.

Generating the Relay Server Outbound Enabler Configuration File

To quickly and easily replicate a common outbound enabler (RSOE) configuration to multiple hosts, generate an RSOE configuration file.

Administrators can use SAP Control Center to configure an initial RSOE for development. Once a configuration proves valid and stable, the administrator can generate the RSOE configuration file, then use `regRelayServer.bat` to apply it to SAP Mobile Server hosts.

1. In the left navigation pane of SAP Control Center, click the SAP Mobile Server cluster name.
2. In the right administration pane, click the **Relay Servers** tab.
3. Select one or more relay server configurations.
4. Click **Generate**.
5. Choose **Outbound enabler configuration XML file**, then click **Next**.
6. Click **Finish**.
7. Select an output target for the file.

See also

- *Configuring Relay Server Properties* on page 72

Generating and Modifying the Relay Server Configuration File

Generate all or part of a Relay Server configuration file. Then transfer the generated file to all Relay Server hosts.

Generating a configuration file extracts the property values stored in the cluster database during the configuration process, and writes them to a file. You may still need to edit this file.

1. In the navigation pane, click the SAP Mobile Server cluster name.
2. In the administration pane, click the **Relay Servers** tab.
3. Select a Relay Server configuration.

4. Click **Generate**.
5. Choose **Relay server configuration file**.
6. Select the parts of the file to generate:
 - The entire Relay Server configuration
 - A server node definition
 - A farm definition
7. Click **Next**, then click **Finish**.
8. Select an output target for the file.
9. Manually edit the file if necessary, and save the changes.
For details on other manual edits that you can perform, see the Relay Server documentation at <http://dcx.sybase.com/index.html#sa160/en/relayserver/relayserver16.html>.
10. To configure a Relay Server farm, apply the same changes to the configurations of remaining farm members. The configuration among all members must be identical.

Setting Up RSOE

Set up one or more RSOEs for each SAP Mobile Server identified in a Relay Server configuration. The configured values are saved in the cluster database.

Configuring RSOE General Properties

Set general RSOE configuration properties to define the context in which the RSOE process operates.

1. In the navigation pane, click **Servers** > <*ServerNode*>.
2. In the administration pane, select the **Outbound Enabler** tab, then click **New**.
3. Specify these property values:
 - **Farm type** – select the type of request managed by the Relay Server: Replication, Messaging or Webservice protocol. When configuring Relay Server Outbound Enabler properties for a scale-out node, you can select only the Webservice farm type.
 - **Mobile Server port** – select the port on which RSOE manages requests.
 - **Relay server host** – if the Relay Server farm has a load balancer in front of it, the host name or IP address of the load balancer. Otherwise, the host name or IP address of a single Relay Server.
 - **Relay server port** – for Relay Server farms that use a load balancer, the port of the load balancer. Otherwise, the Relay Server HTTP or HTTPS port.
 - **Mobile Server farm** – select the string that identifies the SAP Mobile Server cluster, for which the Relay Server manages requests. This property is case-sensitive, and it must match the value in the Relay Server configuration.

- **Server node ID** – select the string that identifies the SAP Mobile Server in the cluster. This property is case-sensitive, and must match the value in the Relay Server configuration.

4. Click **Next**.

Configuring RSOE Connection Settings

Set connection configuration properties for an RSOE. These properties define the RSOE connection to the Relay Server.

1. Configure SSL connection options:

- **Server Authentication** – select this option if you want Relay Server authentication for the SSL connection.
- **Mutual Authentication** – select this option if you want mutual authentication for the SSL connection.
- **Key Alias** – this option is only enabled if you selected mutual authentication. Select an outbound enabled identify key alias whose public certificate is trusted by the Relay Server.

2. Specify these property values:

- **Certificate file** – select this option and choose the .CRT file used to authenticate the RSOE to Relay Server. You can choose this file only if you have already loaded it into the SAP Mobile Server certificate store and your Relay Server Port selection is HTTPS:443 in General Properties.

3. Specify these property values:

- **HTTP user** – select the user name for RSOE authentication on the Web server (Relay Server host).
- **HTTP password** – enter the password for RSOE authentication on the Web server.

4. If RSOE connections to the Relay Server must pass through an Internet proxy server, specify these property values:

- **Proxy server** – select the Internet proxy server.
- **HTTP proxy user** – select the user name for RSOE authentication on the proxy server.
- **HTTP proxy password** – type the password for RSOE authentication on the proxy server.

Configuring RSOE Start Options

Configure start options for RSOE.

1. Enable an option:

- a) Select the box that corresponds to each name.
- b) Set a value.

2. Click **OK**.

3. Ensure the process starts by viewing the Status column of the Outbound Enablers tab.

Managing Configured Relay Servers

Relay Servers configured with SAP Control Center are registered in the SAP Mobile Server cluster database. Administrators can view or edit configuration properties, and delete Relay Servers in SAP Control Center when they are displayed in the **Relay Server** tab.

Viewing or Editing Relay Server Properties

View or edit configuration properties for a selected Relay Server.

1. *Relaunching the Relay Server Configuration Wizard*

Relaunch the Relay Server Configuration wizard to create a new Relay Server configuration file, with customized property values.

2. *Setting Relay Server General Properties*

Set basic connection properties for the Relay Server.

3. *Defining Server Farms and Cluster Nodes*

Set connection properties for the SAP Mobile Server cluster and its constituent nodes.

4. *Reviewing Configured Relay Server Properties*

Confirm the Relay Server property values before you generate the configuration file.

See also

- *Deleting a Relay Server Configuration* on page 83
- *Refreshing the Relay Server List* on page 83

Relaunching the Relay Server Configuration Wizard

Relaunch the Relay Server Configuration wizard to create a new Relay Server configuration file, with customized property values.

1. In the navigation pane, click the SAP Mobile Server cluster name.
2. In the administration pane, click the **Relay Server** tab.
3. Select a Relay Server.
4. Click **Properties**.

Setting Relay Server General Properties

Set basic connection properties for the Relay Server.

Prerequisites

Launch the Relay Server configuration wizard.

Task

1. Select and specify property values for either:

- **Standalone relay server** – outbound enablers bypass the load balancer and high availability is compromised if a direct Relay Server connection fails.
 - **Host** – the host name or IP address of a single Relay Server
 - **HTTP port** – the Relay Server HTTP port
 - **HTTPS port** – the Relay Server HTTPS port

Note: If Relay Server uses HTTPS and certificates, clients other than those using replication-based synchronization may not be able to connect: messaging applications support only HTTP, and Hybrid Web Container applications for iOS support HTTPS, but not certificates.

- **URL suffix** – the URL suffix used by the Outbound Enabler to connect to a Relay Server. The value you set depends on whether Relay Server is installed on IIS or Apache hosts. For IIS, use `/ias_relay_server/server/rs_server.dll`. For Apache use `/srv/iarelayserver/`.

For IIS, the value identifies the *relative* path for `rs_client.dll`. If your IIS directory structure is different, modify this value accordingly.

- **(Optional) Description** – a user-defined description of the Relay Server.
- **Relay server farm** – configures a farm of relay servers, including values for the load balancer that is in front of the Relay Server farm. Values vary for load balanced environments.
 - **Load balancer host** – the host name or IP address of the load balancer in front of the Relay Server farm
 - **HTTP port** – the port of the load balancer
 - **HTTPS port** – the secure port of the load balancer
 - **URL suffix** – the URL suffix used by the Outbound Enabler to connect to a Relay Server load balancer. The value you set depends on whether Relay Server load balancer is installed on IIS or Apache hosts. The default for IIS is `/ias_relay_server/server/rs_server.dll`. The default for Apache is `/srv/iarelayserver/`.

For IIS, the value identifies the *relative* path for `rs_client.dll`. If your IIS directory structure is different, modify this value accordingly.

- **(Optional) Description** – a user-defined description of the Load Balancer.
- **Relay server nodes within farm** – The nodes of relay server within a relay server farm. Add or remove relay server nodes:

To add relay server nodes, specify these property values and click +:

- **Host** – the host name or IP address of a single Relay Server
- **HTTP port** – the Relay Server HTTP port

- **HTTPS port** – the Relay Server HTTPS port

To remove relay server nodes, select the corresponding node, then click **X**.

2. Add or remove HTTP credentials as required:

- a) Select **Configure relay server HTTP credentials**.
- b) To add new credentials, specify these property values and click +:
 - **User name** – user name for RSOE authentication on the Web server (Relay Server host).
 - **Password** – password for RSOE authentication on the Web server.
- c) To remove credentials from the list, select the corresponding user name, then click **X**.

3. Click **Next**.

Defining Server Farms and Cluster Nodes

Set connection properties for the SAP Mobile Server cluster and its constituent nodes.

1. Define the SAP Mobile Server cluster.

- a) Specify these property values:
 - **Farm ID** – a string that identifies the SAP Mobile Server cluster for which the Relay Server manages requests. This property is case-sensitive, and must match the value in the Outbound Enabler configuration.
 - **Type** – the type of request managed by the Relay Server: Replication, Messaging or Webservice protocol. When configuring Relay Server Outbound Enabler properties for a scale-out node, you can select only the Webservice farm type.
 - **(Optional) Description** – user-defined description for the SAP Mobile Server cluster.
- b) Click +.
- c) Repeat steps 1 and 2 to add multiple SAP Mobile Server clusters.
- d) To delete a configured SAP Mobile Server cluster, select it in the list, then click the **X** button.

2. Identify each SAP Mobile Server instance in the cluster.

- a) Select an existing SAP Mobile Server cluster.
- b) Specify these property values:
 - **Node ID** – a string that identifies the SAP Mobile Server in the cluster. This property is case-sensitive, and it must match the value in the RSOE configuration.
 - **Token** – the security token used by the Outbound Enabler to authenticate its connection with the Relay Server. Assign a token string (up to 2048 characters); one token can be shared by all farm types.
- c) Click +.
- d) Repeat steps 1 and 2 to add SAP Mobile Server cluster nodes.

- e) To delete a configured SAP Mobile Server node, select it in the list and click **X**.
3. Click **Next** to review your settings, or click **Finish** to exit the wizard.

The relay server configuration screen displays the Relay Server as Type STANDALONE for a standalone Relay Server or LOAD_BALANCER for a Relay Server farm.

Note: After you exit the wizard, generate the Relay Server configuration file, and copy it to each Relay Server instance to update configuration for multiple Relay Servers.

The Relay Server is registered with SAP Control Center, and can be managed from the Relay Servers tab for the SAP Mobile Server cluster.

Reviewing Configured Relay Server Properties

Confirm the Relay Server property values before you generate the configuration file.

1. Review property values to ensure that:
 - No errors exist.
 - All SAP Mobile Server clusters are defined, and assigned the correct type.
2. Click **Finish**.

The Relay Server is registered with SAP Control Center, and it can be managed from the Relay Servers tab for the SAP Mobile Server cluster.

Deleting a Relay Server Configuration

Delete a Relay Server configuration to remove all defined SAP Mobile Server clusters, server nodes, and RSOEs that connect to the Relay Server.

1. In the navigation pane, click the SAP Mobile Server cluster name.
2. In the administration pane, click the **Relay Server** tab.
3. Select a Relay Server.
4. Click **Delete**.

See also

- *Viewing or Editing Relay Server Properties* on page 80
- *Refreshing the Relay Server List* on page 83

Refreshing the Relay Server List

Refresh the Relay Server list to display current information about deployed and configured Relay Servers.

1. In the navigation pane, click the SAP Mobile Server cluster name.
2. In the administration pane, click the **Relay Server** tab.

3. Click **Refresh**.

See also

- *Viewing or Editing Relay Server Properties* on page 80
- *Deleting a Relay Server Configuration* on page 83

Relay Server Tab Reference

Configuration property values that appear in the Relay Server tab for an SAP Mobile Platform cluster.

Column	Description
Host	if the Relay Server farm has a load balancer in front of it, the host name or IP address of the load balancer. Otherwise, the host name or IP address of a single Relay Server.
HTTP port	if the Relay Server farm has a load balancer in front of it, the port of the load balancer. Otherwise, the Relay Server HTTP port.
HTTPS port	if the Relay Server farm has a load balancer in front of it, the port of the load balancer. Otherwise, the Relay Server HTTPS port.
URL suffix	the URL suffix used by the Outbound Enabler to connect to a Relay Server. The value you set depends on whether Relay Server is installed on IIS or Apache hosts. For IIS, use <code>/ias_relay_server/server/rs_server.dll</code> . For Apache use <code>/srv/iarelayserver/</code> .
Type	STANDALONE for a standalone Relay Server or LOAD_BALANCER for a Relay Server farm.
(Optional) Description	a user-defined description of the Relay Server.

SAP Mobile Server

The SAP Mobile Platform runtime server is called SAP Mobile Server. SAP Mobile Server is an instance of SAP Mobile Server installed as the application server to manage resource intensive traffic/communication/transactions, such as PUSH payload and synchronization, between the scale-out nodes and the data server cluster. SAP Mobile Server can be installed as a cluster. In a production environment, the SAP Mobile Server must be installed on a 64-bit host.

There are two types of server nodes that can be installed:

- Application Service node – (mandatory) runs all services.

- Scale Out node – (optional) specifically designed to allow the stateless request/response HTTP and synchronous message services to be horizontally scaled.

SAP Mobile Server features include:

- Data services – supports connections to back-end data resources using these standard technologies: enterprise databases with JDBC™ connections and Web Services (SOAP-style and REST-style) . Also supports connections to enterprise applications such as SAP®.
- Data virtualization – introduces a layer called a mobile business object (MBO) between your enterprise databases or applications, and the remote database on the device client. Utilizes a cache database (CDB) to optimize device client access and minimize back-end resource utilization.
- Device connection services – (does not apply to Scale Out nodes) supports connections from various different platforms and operating systems with different communication styles.
 - Replication-based synchronization – A synchronization method where cached data is downloaded to and uploaded from client database to server via replication. Typically, mobile replication-based synchronization is used in occasionally connected scenarios.
 - Messaging-based synchronization – In flight messages are queued in a messaging cache. Synchronization occurs as messages are delivered to the device. Typically, mobile messaging-based synchronization is used in always available and occasionally disconnected scenarios.

See also

- *Server List* on page 85
- *Cluster Properties* on page 41
- *Server Log* on page 96

Server List

Depending on the license you purchase and the type of environment you install, you may deploy multiple SAP Mobile Servers in a cluster.

There are two types of server nodes that can be installed:

- Application Service node – (mandatory) runs all services.
- Scale Out node – (optional) specifically designed to allow the stateless request/response HTTP and synchronous message services to be horizontally scaled.

If you have installed multiple servers as part of a clustered architecture, you must register these servers first. Only servers that are installed on the same host as SAP Control Center are registered automatically. Once registered, remote servers also appear in the server list.

Servers are listed according to their cluster role (that is, primary or secondary servers). SAP Control Center automatically identifies the primary server and lists it first, followed by secondary servers.

The **Server Type** column indicates whether the server is an Application Server node or a Scale Out node.

The **Need Restart** column indicates if a server restart is required due to configuration changes.

See also

- *SAP Mobile Server* on page 84
- *Cluster Properties* on page 41
- *Server Log* on page 96

Stopping and Starting a Server

Stop and start a server to perform maintenance or to apply changes to server settings. You can perform this action as a two-step process (stop and start) or as a single restart process.

You can stop and start a server from SAP Control Center for servers that are installed on the same host as SAP Control Center, as well as servers that are installed on different hosts.

Note: If someone manually shuts the server down, this action triggers multiple errors in SAP Control Center SAP Mobile Server until the console determines that the server is no longer available. This takes approximately 30 seconds to detect. When this occurs you might see multiple `Runtime API throws exception` errors logged. Wait for the server to come online and log into the server again to resume your administration work.

Note: SAP Control Center requires at least one Application Server node running to work properly. If all Application Server nodes are stopped, the SAP Control Center console tree will be collapsed automatically and the following error message will display: `The cluster is unavailable because of no running primary server found.`

Note: You cannot start a Scale-out node from SAP Control Center. If you stop a Scale-out node, you must start it manually.

1. In the SAP Control Center navigation pane, click **Servers** to display the servers list.
2. Select a server in this list.
3. Choose an appropriate process:
 - To stop the server, click **Stop**. You can then perform the administration actions you require that might require the server to be started. To then restart the server, click **Start**.

Note: If you have selected a Scale-out node server type, the **Start** button is disabled.

- If you perform an administration action that requires a restart to take effect, click **Restart**. This shuts the server down and restarts it in a single process.

As the server stops and starts, progress messages display in the Server Console pane.

Suspending and Resuming a Server

Suspend and resume a server to temporarily disallow clients to access the specific server for routine maintenance. While the server is suspended, it remains running and available for all administrative actions.

Prerequisites

Configure the Relay Server Outbound Enabler (RSOE) for SAP Mobile Server in order to enable the suspend and resume server functions.

Task

1. In the SAP Control Center navigation pane, click **Servers** to display the servers list.
2. Select a server in the list.
3. Choose an appropriate process:
 - To suspend the server, click **Suspend**. Wait for about 1 minute, and click **Refresh**. The **Suspend** button is now disabled. You can perform the administration actions you require.
 - To resume the server, click **Resume**.

As the server suspends and resumes, progress messages display in the Server Console pane.

Pinging a Server

Ping a server to test the availability of backend server connectivity and verify the server state (for example, started or stopped). By default ping uses whichever Internet Inter-ORB Protocol call you configured (IIOPS by default) to test if a server's connection is available.

1. In the left navigation pane, expand the **Servers** folder and select a server.
2. Select the **General** tab.
3. Click **Ping**.

The result displays in the console area.

Checking SAP Mobile Server Status

Verify whether a server is running, stopped, or suspended.

1. In the left navigation pane, select **Servers**.
2. In the right administration pane, select the **General** tab.
3. In the Status column, check the server status corresponding to the server you are administering: running or stopped.
4. Use the controls in the administration console to start, stop, or restart the server, as required.

Resetting JCo Repository Caches for a Server

Reset the JCo repository caches for a server to reduce the amount of heap memory used by the repository caches.

1. In the left navigation pane, expand the **Servers** folder and select a server.
2. Select the **General** tab.
3. Click **Reset JCo Repository Caches**.
4. Click **Yes** to confirm the operation.

The result displays in the message area.

Relay Server Outbound Enabler

The Outbound Enabler (RSOE) runs as an SAP Mobile Server process and manages communication between the SAP Mobile Server and a Relay Server.

Each RSOE maintains connections to each Relay Server in a Relay Server farm. The RSOE passes client requests to the SAP Mobile Server on its Replication or Messaging port. The SAP Mobile Server sends its response to the RSOE, which forwards it to the Relay Server, to be passed to the client.

As an SAP Mobile Server process, the RSOE always starts when SAP Mobile Server starts. SAP Mobile Server monitors the process to ensure it is available. If an RSOE fails for any reason, SAP Mobile Server restarts it automatically.

Note: SAP recommends three RSOE processes each, for both Replication and Messaging ports.

Loading and Unloading HTTPS Certificates for RSOE

Load HTTPS certificates for the RSOE to add it to the SAP Mobile Server node .

Note: You must use only RSA certificates.

If the Web server (Relay Server host) already uses a certificate signed by a CA for HTTPS connections, you do not need to perform this task.

1. In the navigation pane, click **Servers > ServerNode**.
2. In the administration pane, select the Outbound Enabler tab, then click **Certificate Files**.
3. Choose the action you want to perform:
 - To add a new certificate, click +. Browse and select the .CRT file to upload, then click **Open**.
 - To replace a certificate in the store, select **Replace the certificate file**. Verify the certificate file name, then click +.

- To delete a certificate from the store, select the filename and click **X**.
4. When certificate management tasks are complete, click **OK**.

Setting Up RSOE

Set up one or more RSOEs for each SAP Mobile Server identified in a Relay Server configuration. The configured values are saved in the cluster database.

1. *Configuring RSOE General Properties*

Set general RSOE configuration properties to define the context in which the RSOE process operates.

2. *Configuring RSOE Connection Settings*

Set connection configuration properties for an RSOE. These properties define the RSOE connection to the Relay Server.

3. *Configuring RSOE Start Options*

Configure start options for RSOE.

Configuring RSOE General Properties

Set general RSOE configuration properties to define the context in which the RSOE process operates.

1. In the navigation pane, click **Servers** > <**ServerNode**>.
2. In the administration pane, select the **Outbound Enabler** tab, then click **New**.
3. Specify these property values:
 - **Farm type** – select the type of request managed by the Relay Server: Replication, Messaging or Webservice protocol. When configuring Relay Server Outbound Enabler properties for a scale-out node, you can select only the Webservice farm type.
 - **Mobile Server port** – select the port on which RSOE manages requests.
 - **Relay server host** – if the Relay Server farm has a load balancer in front of it, the host name or IP address of the load balancer. Otherwise, the host name or IP address of a single Relay Server.
 - **Relay server port** – for Relay Server farms that use a load balancer, the port of the load balancer. Otherwise, the Relay Server HTTP or HTTPS port.
 - **Mobile Server farm** – select the string that identifies the SAP Mobile Server cluster, for which the Relay Server manages requests. This property is case-sensitive, and it must match the value in the Relay Server configuration.
 - **Server node ID** – select the string that identifies the SAP Mobile Server in the cluster. This property is case-sensitive, and must match the value in the Relay Server configuration.
4. Click **Next**.

Configuring RSOE Connection Settings

Set connection configuration properties for an RSOE. These properties define the RSOE connection to the Relay Server.

1. Configure SSL connection options:

- **Server Authentication** – select this option if you want Relay Server authentication for the SSL connection.
- **Mutual Authentication** – select this option if you want mutual authentication for the SSL connection.
- **Key Alias** – this option is only enabled if you selected mutual authentication. Select an outbound enabled identify key alias whose public certificate is trusted by the Relay Server.

2. Specify these property values:

- **Certificate file** – select this option and choose the .CRT file used to authenticate the RSOE to Relay Server. You can choose this file only if you have already loaded it into the SAP Mobile Server certificate store and your Relay Server Port selection is HTTPS:443 in General Properties.

3. Specify these property values:

- **HTTP user** – select the user name for RSOE authentication on the Web server (Relay Server host).
- **HTTP password** – enter the password for RSOE authentication on the Web server.

4. If RSOE connections to the Relay Server must pass through an Internet proxy server, specify these property values:

- **Proxy server** – select the Internet proxy server.
- **HTTP proxy user** – select the user name for RSOE authentication on the proxy server.
- **HTTP proxy password** – type the password for RSOE authentication on the proxy server.

See also

- *Updating Common Properties for Multiple RSOEs Concurrently* on page 92
- *Outbound Enabler Start Options Reference* on page 91

Configuring RSOE Start Options

Configure start options for RSOE.

1. Enable an option:

- a) Select the box that corresponds to each name.
- b) Set a value.

2. Click **OK**.

3. Ensure the process starts by viewing the Status column of the Outbound Enablers tab.

Outbound Enabler Start Options Reference

Review available outbound enabler start options, which affect outbound enabler logging. Each outbound enabler has its own log file that you can retrieve in SAP Control Center.

Option	Default	Description
Verbosity level	0	Sets log file verbosity values: • 0 – log errors only. Use this logging level for deployment. • 1 – session-level logging. This is a higher level view of a session. • 2 – request-level logging. Provides a more detailed view of HTTP requests within a session. • 3 - 5 – detailed logging. Used primarily by Technical Support.
Reconnect delay	5	Delay before retry after connection fails.
Maximum output file size	10KB	Maximum log file size.
Truncate log file	None	Delete the log file at RSOE startup.
Advanced	None	User-defined value for start parameters. See <i>Outbound Enabler</i> in <i>SQL Anywhere 12.0.1</i> at http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.help.sqlanywhere.12.0.1/relayserver/ml-relayserver-s-6039420.html .

See also

- *Updating Common Properties for Multiple RSOEs Concurrently* on page 92
- *Configuring RSOE Connection Settings* on page 90

Generating the Relay Server Outbound Enabler Configuration File

To quickly and easily replicate a common outbound enabler (RSOE) configuration to multiple hosts, generate an RSOE configuration file.

Administrators can use SAP Control Center to configure an initial RSOE for development. Once a configuration proves valid and stable, the administrator can generate the RSOE configuration file, then use `regRelayServer.bat` to apply it to SAP Mobile Server hosts.

1. In the left navigation pane of SAP Control Center, click the SAP Mobile Server cluster name.
2. In the right administration pane, click the **Relay Servers** tab.
3. Select one or more relay server configurations.
4. Click **Generate**.
5. Choose **Outbound enabler configuration XML file**, then click **Next**.
6. Click **Finish**.
7. Select an output target for the file.

Managing Configured RSOEs

Manage RSOE instances you have configured.

Retrieving RSOE Logs

You can retrieve one RSOE log at a time, from the SAP Mobile Server host, and copy it to another location. You cannot retrieve an empty RSOE log file.

1. In the navigation pane, click **Servers** > <**ServerNode**>.
2. In the administration pane, select the Outbound Enabler tab, and select the RSOE instance.
3. Click **Retrieve Log**, then **Next**, then **Finish** to save the log and choose the target location for the file.

See also

- *Updating Common Properties for Multiple RSOEs Concurrently* on page 92
- *Configuring Proxy Server Settings for an Outbound Enabler* on page 93

Updating Common Properties for Multiple RSOEs Concurrently

To avoid setting common RSOE properties repeatedly, configure common properties simultaneously for selected RSOEs already deployed. This streamlines the number of times they would otherwise need to be set.

1. In the navigation pane, click **Servers** > <**ServerNode**>.
2. In the administration pane, select the Outbound Enabler tab.

3. Select two or more RSOE instances in the list of configured RSOEs, then click **Properties**.
4. Modify common settings for multiple RSOEs, including:
 - Startup options. See *Outbound Enabler Start Options Reference*.
 - Proxy settings. See step 2 in *Configuring RSOE Connection Settings*.

See also

- *Retrieving RSOE Logs* on page 92
- *Configuring Proxy Server Settings for an Outbound Enabler* on page 93
- *Configuring RSOE Connection Settings* on page 90
- *Outbound Enabler Start Options Reference* on page 91

Configuring Proxy Server Settings for an Outbound Enabler

(Applies only to Online Data Proxy) Configure an Outbound Enabler to work with an Internet proxy server, when connections to the Relay Server must pass through a proxy server.

1. In the navigation pane, click **Servers**><**ServerNode**> .
2. In the administration pane, select the **Outbound Enabler** tab.
3. Click **Proxy** .
4. Define a list of required proxy servers:
 - a) To add a new server connection, type **Host** and **Port** values, then click +.
 - b) To remove an existing connection, select the server, then click **X**.
 - c) To edit an existing connection, click an appropriate cell and re-enter or modify the current value.
5. Define a proxy user for a selected server:
 - a) Select a server from the list.
 - b) To add a new user, enter a User name and password then click +.
 - c) To remove an existing user, select the name, then click **X**.
 - d) To edit an existing user, click an appropriate cell and re-enter or modify the current value.
6. Click **OK**.

See also

- *Retrieving RSOE Logs* on page 92
- *Updating Common Properties for Multiple RSOEs Concurrently* on page 92

Outbound Enabler Tab Reference

Understand the columns of data displayed in the Outbound Enabler tab for an SAP Mobile Server node.

Column Name	Displays
Server Node ID	the string that identifies the SAP Mobile Server in the cluster. This property is case-sensitive, and must match the value in the Relay Server configuration.
Mobile Server Port	the port on which Outbound Enabler manages requests.
Farm Type	the type of request managed by the Relay Server: Replication, Messaging or Webservice protocol. When configuring Relay Server Outbound Enabler properties for a scale-out node, you can select only the Webservice farm type.
Mobile Server Farm	the string that identifies the SAP Mobile Server cluster, for which the Relay Server manages requests. This property is case-sensitive, and it must match the value in the Relay Server configuration.
Relay Server Host	if the Relay Server farm has a load balancer in front of it, the host name or IP address of the load balancer. Otherwise, the host name or IP address of a single Relay Server.
Status	current state of the Outbound Enabler process: stopped, running, or error.

Column Name	Displays
Status Description	additional details on the state of the Outbound Enabler. If you receive one of these messages, follow the documented recommendation: • Unknown error state – Check the log for additional details. • Failed to connect SAP Mobile Server, retrying... – Check the SAP Mobile Server port managed by the Outbound Enabler. • Unauthorized. – Check the security token of Outbound Enabler. • Unrecognized farm or server node ID. – The string that identifies the SAP Mobile Server cluster or server node in the Outbound Enabler configuration does not match the Relay Server configuration. • Please check the relay server host and port or Failed to create I/O stream to the relay server – If you use HTTPS port, check to see if the certificate file is valid. • Relay server service unavailable. – Check if the Relay Server is properly configured, or if any internal errors are logged. • Relay server not found. – Either the Relay Server is not yet deployed, or the URL suffix is wrong. • Bad request. – Check the syntax of the URL suffix. • Error writing HTTP headers – Check if the trusted certificate is valid, and verify the URL suffix syntax. Something may be misformatted. Note: Sometimes when the Status column shows "Running" the Status Description shows: <pre>Relay Server outbound enabler is running. Please check the log file to confirm the status.</pre> In these cases, the console may detect that an RSOE is running, even though the RSOE is actually in an error state. • The RSOE log level is set too high (4 or 5). SAP Control Center cannot detect the status from scanning the RSOE log. • The RSOE enters an unrecognized error condition. For example, when RSOE connects to Relay Server through an Internet proxy server, if the proxy server shuts down, the RSOE is effectively in an error state. The RSOE may continue to retry the connection indefinitely, and produce no log message recognized as an error.
Certificate File	the certificate file uploaded to the SAP Mobile Server certificate store.

Column Name	Displays
Log File	the name and location of the Outbound Enabler log file. The syntax of this filename is <code><nodeName>.RSOE<n>.log</code> . <code><n></code> is the primary key of the Outbound Enabler database record in the cluster database used by SAP Mobile Platform.

Server Log

Server logs enable you to monitor system health at a high level, or focus in on specific issues by setting up filtering criteria using SAP Control Center.

These server logs are available:

- SAP Mobile Server logs – collect data on SAP Mobile Server health and performance by component, and retrieve data for all or specific searches. You can save and archive system logs.
- Messaging Server logs – retrieve trace data for all or specific messages. Export data for archive or further analysis.

See also

- *Server List* on page 85
- *SAP Mobile Server* on page 84
- *Cluster Properties* on page 41

SAP Mobile Server Runtime Logging

SAP Mobile Server logs collect runtime information from various embedded runtime components.

You can change default log levels for different components as required from SAP Control Center.

You can view logs from:

- SAP Control Center – click **Servers** > **primaryServer** > **Log** in the left pane.
The first 150 entries initially appear in the console area, so you may need to incrementally retrieve more of the log as required, by scrolling down through the log events.
- Text editor – browse to and open one or more of the `SMP_HOME\Servers\UnwiredServer\logs\<hostname>-server.log` files. These files may be indexed, depending on how you configure the life cycle for the server's log file.

See also

- *Messaging Server Runtime Logging* on page 98

Viewing the SAP Mobile Server Log

In text or grid view, use the vertical scroll bar to retrieve additional segments of the log file in 150-line increments. In grid view, up to 10 pages of the server log data is loaded in one request.

You can navigate to any page by using the **First**, **Prev**, **Next**, and **Go to** controls. Use **View Details** open the actual log file and find the corresponding line.

Note: **Last** is disabled -- use **Next** to advance to the final page in the list.

There are also two search options you can use:

- Basic search – search by keyword, log level, first/last X number of lines in the log file.
- Advanced search – search by specific subcomponents, log level, exception, time range, and so on.

You can include backup logs in your search or retrieval. The option is not selected by default.

Searching SAP Mobile Server Log Data

Filter server log data according to the criteria you specify.

1. In the SAP Control Center left navigation pane, click **Servers > primaryServer > Log**, and in the right pane click **SAP Mobile Server**.
2. Select **Show filter criteria** to display the search pane.
3. Select **Include backup logs** to display backup logs.
4. Select **Text view** or **Grid view** to specify how to display the logs.
5. Select **Basic search** to filter your search according to the specific string you enter in the search field. (Optional) You may also specify:
 - **Show** – specify first lines, last lines, or a keyword. If you are searching by first or last lines, you can enter any value up to a maximum of 1000 lines in the log. However, SAP recommends that you provide a more manageable value to avoid severe performance degradation associated with this upper limit.
 - **Log level** – search only messages logged by the particular log level you select.
6. Select **Advanced search** to enter more specific search criteria, including:
 - **Component** – identify which component the log data belongs to: MMS, Proxy, MSG, Security, MobiLink™, DataServices, Other or DOEC.

Note: Set the log level for each component in the **Setting** tab. See *Configuring SAP Mobile Server Log Settings*.

- **Log level** – search only messages logged by the particular log level you select.
- **Thread ID** – specify the ID name of the thread that logs the message you are searching.
- **Logger name** – indicate the class name and instance of the logged component.
- **Keyword** – indicate a value, file name, or other keyword by which to filter your search.

- **Time period** – specify a start date, start time, end date, and end time.
7. Click **Retrieve**.
 8. To begin a new query, click **Reset** in the search panel and enter new search criteria.

Retrieving the SAP Mobile Server Log

Update the information in the log console window.

1. In the SAP Control Center left navigation pane, click **Servers** > **primaryServer** > **Log**, and in the right pane click **SAP Mobile Server**.
2. To display the latest log data collected in the log file if time has elapsed since you last opened the log, click **Retrieve**.
3. (Optional) Select a row to view a single record in the detail pane. Additional columns may be available.

Deleting the SAP Mobile Server Log

Clear old or unrequired server log data from the log file.

1. In the SAP Control Center left navigation pane, click **Servers** > **primaryServer** > **Log** and in the right pane click **Mobile Server**.
2. To delete all data from the log file and all backup log files, click **Delete**, then **OK**.

Messaging Server Runtime Logging

(Does not apply to Scale Out server nodes) Messaging Server logs collect data that enables you to trace message handling from the cluster database to the device user, based on various trace settings.

See also

- *SAP Mobile Server Runtime Logging* on page 96

Viewing the Messaging Server Log

You can view results for one or more modules, or the Default. You can navigate to any page by using the **First**, **Prev**, **Next**, **Last**, and **Go to** controls.

Searching Messaging Server Log Data

Filter server log data according to the criteria you specify.

1. In the SAP Control Center left navigation pane, click **Servers** > **primaryServer** > **Log**, and in the right pane click **Messaging Server**.
2. Click **Show filter**, and then select the search criteria:
 - **Max level** – search only messages logged by the particular log level you select. All messages up to that level are retrieved.

- **Thread ID** – specify the ID name of the thread that logs the message you are searching.
- **Contains** – enter a search string.
- **Users** – select one or more users.
- **Application connections** – select one or more application connections.
- **Modules** – select one or more modules.
- **Time period** – specify a start date, start time, end date, and end time.

3. Click **Retrieve**.

4. To begin a new query, click **Reset**.

Retrieving the Messaging Server Log

Update the information in the log console window.

1. In the SAP Control Center left navigation pane, click **Servers > primaryServer > Log**, and in the right pane click **Messaging Server**.
2. To display the latest log data collected in the log file if time has elapsed since you last opened the log, click **Retrieve**.
3. (Optional) Select a row to view a single record in the detail pane. Additional columns may be available.

Exporting Messaging Server Log

Export retrieved trace information for archive or further analysis.

1. In the SAP Control Center left navigation pane, click **Servers > primaryServer > Log**, and in the right pane click **Messaging Server**.
2. To display the latest log data collected in the log file if time has elapsed since you last opened the log, click **Retrieve**.
3. Click **Export** to launch the Export Trace Log Wizard.

Trace Log

The trace logs capture messaging server data for cluster level database to mobile device user activities. Using the trace logs you can trace obtain detailed information using a variety of search criteria.

- **Time** – the date and time when the current trace entry was logged on SAP Mobile Server. The returned date and time is the SAP Mobile Server time without time zone information.
- **Module** – the module to which the current trace entry belongs.
- **Description** – detailed trace information.
- **Level** – the trace level of the current trace entry. The possible trace level values (from high to low) are: ERROR, WARN, INFO, and DEBUG.
- **User** – the user name of the current trace entry.
- **Application Connection ID** – the application connection ID of the current trace entry.

- Thread ID – the thread ID when the trace entry was logged.
- Node – the server that created the trace entry.

Domains

Domains provide a logical partitioning of a hosting organization's environment that achieves increased flexibility and granularity of control in multitenant environments. By default, the installer creates a single domain named "default."

Administrators use different domains within the same SAP Mobile Platform installation. Domains enable the management of application metadata within a partition, including server connections, packages, role mappings, domain logs, and security, so that changes are visible only in the specific domain.

Considerations when implementing domains in a multitenant environment include:

- Create and manage domains using SAP Control Center from the SAP Mobile Platform administration perspective of SAP Control Center.
- You can support multiple customers inside the same SAP Mobile Platform cluster.
- You can configure security specifically for individual domains by creating one or more security configurations in the cluster, and then assigning those security configurations to a domain. You can then map the security configurations to one or more packages. A user accessing the package from a device application is authenticated and authorized by the security provider associated with the package.
- Customers may require their own administrative view on their portion of the SAP Mobile Platform-enabled mobility system. By granting domain administration access to your customers, you can allow customers to customize their deployed applications packages and perform self-administration tasks as needed.

The "default" domain

The "default" domain is a special domain where critical runtime configuration artifacts exist. These artifacts include:

- An "admin" security configuration – this security configuration is mapped to the "default" domain and is used to authenticate and authorize administrative users. For this reason, administrators are not allowed to unassign the "admin" security configuration from the "default" domain.
- Cache database (CDB) data source connections – for the "default" CDB data source, users can configure the Pool Size property in the "default" domain according to their requirements. This setting allows the maximum number of open connections to the SQL Anywhere database server hosting the CDB.
- Monitor database data source connections – the customer can modify the existing monitoring data source properties according to their configuration requirements, or create a new monitoring datasource in the "default" domain.

- Domain log database data source connections – the customer can modify the existing domain log data source properties according to their configuration requirement, or create a new domain log data source in the "default" domain. By default, the name of domain log data source is "domainlogdb".

Since these critical runtime-related artifacts are located in the "default" domain, administrators are not allowed to delete this domain. SAP recommends creating new domains to facilitate tenants according to their application requirements.

Creating and Enabling a New Domain

Create and configure multiple domains within a single SAP Mobile Platform installation. A domain must be enabled for application users to access the packages deployed in the domain. Enabling a domain also triggers synchronization of the domain changes to the secondary nodes in the cluster. Application users who attempt to access a disabled domain receive an error message.

Prerequisites

Create a security configuration for the domain and register the domain administrator.

Task

1. In the left navigation pane, select the **Domains** folder.
2. In the right administration pane, select the **General** tab, and click **New**.
3. In the Create Domain dialog, enter a name for the domain and click **Next**.

Note: Domain names are case-insensitive.

4. Select a security configuration for the domain by checking an option from the list of available configurations. You must select at least one security configuration. The security configurations you select are then available for use in validating users accessing the packages. If you select multiple security configurations, the first one you select becomes the default security configuration for the domain.
5. Click **Next**.
6. Optional. Select one or more domain administrators for the domain.
7. Click **Finish**.
The new domain appears in the **General** tab.
8. Click the box adjacent to the domain name, click **Enable**, then click **Yes** to confirm.

See also

- *Deleting a Domain* on page 102
- *Registering a Domain Administrator User* on page 103
- *Assigning Domain Administrators to a Domain* on page 104
- *Viewing Applications for a Domain* on page 105

- *Viewing Application Connections for a Domain* on page 105
- *Scheduling Accumulated Data Cleanup for Domains* on page 107
- *Domain Logs* on page 110
- *Client Logs* on page 162
- *Connections* on page 167
- *Configuring Domain Security* on page 195

Deleting a Domain

Remove a domain and its contents from the cluster when you no longer require the partition.

When a domain is deleted, all referenced artifacts, such as domain administrators and security configurations, are retained. However, all contained artifacts, including packages, subscription templates, device subscriptions, MBO and operation historical data, package-level role mapping, cache group settings, server connections, and domain-level role mappings for security configurations independent of any other domain, are also deleted.

To preserve a deployed package before deleting a domain, export the package to an archive file.

Note: You cannot delete the "default" domain since it contains critical runtime-related artifacts.

1. In the left navigation pane, select **Domains**.
2. In the right administration pane, click the **General** tab and select the domain you want to delete.
3. Click **Delete**.
4. In the confirmation dialog, click **Yes**.

See also

- *Creating and Enabling a New Domain* on page 101
- *Registering a Domain Administrator User* on page 103
- *Assigning Domain Administrators to a Domain* on page 104
- *Viewing Applications for a Domain* on page 105
- *Viewing Application Connections for a Domain* on page 105
- *Scheduling Accumulated Data Cleanup for Domains* on page 107
- *Domain Logs* on page 110
- *Client Logs* on page 162
- *Connections* on page 167
- *Configuring Domain Security* on page 195

Registering a Domain Administrator User

A platform administrator can add domain administrators, so these users can administer domains to which they are assigned. This process registers an administrator with the cluster, so the user can be assigned as an administrator for a domain.

Prerequisites

Create the user entry and map the physical role to the SUP Domain Administrator logical role in the security provider repository used to authenticate administrators in SAP Control Center (SCC).

Task

1. In the left navigation pane, click the **Security** node.
2. In the right administration pane, click the **Domain Administrators** tab and click **New**.
3. To configure user properties for the administrator, enter:
 - **Login name** – the user name assigned to the administrator. For example, if you are using LDAP to authenticate administrators, the UID is typically used as the login name.

Note: The login name can't be longer than 36 characters.

 - (Optional) **Company name** – the name of the organization the administrator belongs to. SAP recommends you supply this information if you are setting up SAP Mobile Platform in a hosted environment and using domains to distinguish between different hosted solutions for different organizations.
 - (Optional) **First name** – the administrator's first name. The first name must match the one assigned to the login name in the security repository.
 - (Optional) **Last name** – the administrator's last name. The last name must match the one assigned to the login name in the security repository.
4. Click **OK** to register the administrator.
The domain administrator can now log in with his or her user login credentials (user name and password).

Next

Assign the domain administrator role to this user.

See also

- *Creating and Enabling a New Domain* on page 101
- *Deleting a Domain* on page 102
- *Assigning Domain Administrators to a Domain* on page 104
- *Viewing Applications for a Domain* on page 105
- *Viewing Application Connections for a Domain* on page 105

- *Scheduling Accumulated Data Cleanup for Domains* on page 107
- *Domain Logs* on page 110
- *Client Logs* on page 162
- *Connections* on page 167
- *Configuring Domain Security* on page 195

Assigning Domain Administrators to a Domain

Assign domain administration privileges to a domain administrator. You must be a platform administrator to assign and unassign domain administrators.

Prerequisites

Ensure the user is already registered as a domain administrator in the Domain Administrators tab.

Task

1. In the left navigation pane, expand the **Domains** folder, and select the domain for which to assign domain administration privileges.
2. Select the domain-level **Security** folder.
3. In the right administration pane, select the **Domain Administrators** tab, and click **Assign**.
4. Select one or more administrator users to assign to the domain by checking the box adjacent to the user name.
5. Click **OK**.

A message appears above the right administration pane menu indicating the success or failure of the assignment. If successful, the new domain administrator appears in the list of users.

See also

- *Creating and Enabling a New Domain* on page 101
- *Deleting a Domain* on page 102
- *Registering a Domain Administrator User* on page 103
- *Viewing Applications for a Domain* on page 105
- *Viewing Application Connections for a Domain* on page 105
- *Scheduling Accumulated Data Cleanup for Domains* on page 107
- *Domain Logs* on page 110
- *Client Logs* on page 162
- *Connections* on page 167
- *Configuring Domain Security* on page 195

Viewing Applications for a Domain

View applications registered for a specific domain.

1. In the left navigation pane, expand the **Domains** folder, and select a domain.
2. Within the domain, select **Applications**.
3. In the right pane, select the domain-level **Applications** tab.
4. Click **Refresh** to view a list of Applications IDs, and their display names and descriptions.
5. Alternatively, search for one or more application IDs.
 - a) Provide the search criteria for **Application ID** by adding a search string.
 - b) Click **Go**.
All the applications that match the search criteria provided for the selected domain are populated in the table.

See also

- *Creating and Enabling a New Domain* on page 101
- *Deleting a Domain* on page 102
- *Registering a Domain Administrator User* on page 103
- *Assigning Domain Administrators to a Domain* on page 104
- *Viewing Application Connections for a Domain* on page 105
- *Scheduling Accumulated Data Cleanup for Domains* on page 107
- *Domain Logs* on page 110
- *Client Logs* on page 162
- *Connections* on page 167
- *Configuring Domain Security* on page 195

Viewing Application Connections for a Domain

View application connection information for a specific domain. Optionally select the relevant columns of information to display.

Note: The association of application connections to a domain is based on the "domain" setting value in the application connections. Therefore, when registering application connections, the application template must be registered either using a template, where the domain value is appropriately configured, or in the registration wizard of the SAP Control Center user interface.

1. In the left navigation pane, expand the **Domains** folder, and select a domain.
2. Within the domain, select **Applications**.
3. In the right pane, select the domain-level **Application Connections** tab.

4. Click **Refresh** to view a list of Users.
5. Alternatively, search for one or more users.
 - a) Provide the search criteria for **Users** by adding a search string.
 - b) Click **Go**.

All the users that match the search criteria provided for the selected domain are populated in the table.
6. (Optional) Select the columns to display (all columns, or specific columns) from the drop-down list.

See also

- *Creating and Enabling a New Domain* on page 101
- *Deleting a Domain* on page 102
- *Registering a Domain Administrator User* on page 103
- *Assigning Domain Administrators to a Domain* on page 104
- *Viewing Applications for a Domain* on page 105
- *Scheduling Accumulated Data Cleanup for Domains* on page 107
- *Domain Logs* on page 110
- *Client Logs* on page 162
- *Connections* on page 167
- *Configuring Domain Security* on page 195

Managing Connection Pools for SAP Mobile Server Connections

Connection pools are used by SAP Mobile Server to improve performance between the server and internal databases, which include the cache database, domain log database, messaging database, monitor database and sample database. These values are synchronized among all servers in the cluster when the primary server values change.

Configure the maximum pool size to determine the how large of a pool is used for these JDBC database connections.

1. In the left navigation pane, expand the **Domains** folder, and select the domain for which you want to modify the connection.
2. Select **Connections**.
3. In the right administration pane:
 - To edit the properties of a connection pool, click the **Connections** tab.
 - To edit the properties of a connection pool template, click the **Templates** tab.
4. Select a connection pool or template from the list.
5. Select **JDBC** as the **Connection pool type**, and click **Properties**.
6. Change the Max Pool Size value (and any other values you choose). The default value will be different for each database. A value of 0 indicates no limit. The Max Pool Size value should not be a negative value.

See *Creating Connections and Connection Templates* in *SAP Control Center for SAP Mobile Platform* and *JDBC Properties*.

- Click **Save** to save the changes.

Scheduling Accumulated Data Cleanup for Domains

Periodically clean up accumulated data maintenance items in cache that are no longer needed, by creating a data purge schedule. Set the schedule to perform the clean up processes when system usage is low.

Alternatively, manually purge accumulated data items at any time; however, if the number of domain log records is very large (for example, hundreds of thousands) then SAP Mobile Server must purge domain log data asynchronously.

Note: You can use the Administration API to automate cleanup at the package level.

- In the SAP Control Center left navigation pane, expand the **Domains** tab and select a domain.
- In the right pane, select the **Scheduled Tasks** tab.
- Under Task, select one of the options you want to schedule, and then select **Properties** to set up its automatic schedule:

Option	Description
Subscription Cleanup	Removes subscriptions that are inactive for the number of days specified in the schedule task configuration. A subscription is considered active based on: - Replication – last synchronization request time-stamp. - Messaging – last synchronization message time-stamp. A user who infrequently accesses the system, for example, 3-4 times a year, falls outside the specified time frame and is removed from SAP Mobile Platform. He or she must then reinstall the application and initiate a sync to reactivate the subscription.
Error History Cleanup	Removes historical data on MBO data refresh and operation replay failures, which result from system or application failures. System failures may include network problems, credential issues, and back-end system failure. Application failures may include invalid values, and non-unique data. Only error messages are removed.

Option	Description
Client Log Cleanup	Removes logically deleted client log records that have already been synchronized to the device, or are no longer associated with active users.
Synchronization Cache Cleanup	This cleanup task removes: - Logically deleted rows in the cache that are older than the oldest synchronization time on record in the system. Synchronization activity for all clients establish the oldest synchronization time. Note: The last synchronization time of a client is the last time the client successfully applied the download. Because the SAP Mobile Server does not require download acknowledgement, the SAP Mobile Server only knows that the previous download was applied successfully when the client synchronizes for the second time (uploading the previous synchronization time). The synchronization time on record in the system is normally the time the 'second last' synchronization happened. - Unused or stale partitions. In addition, the optional exp header property in DCN messages indicates an expiration date, after which the data is purged based on the defined Synchronization Cache Cleanup schedule.

4. Click **Properties**, then in the Task Properties dialog, select the **Schedule** tab.
5. Configure the required schedule:
 - **Schedule repeat** – select how often the schedule should run. Options are **daily**, **hourly**, **custom**, and **never**.
 - If you select **daily** or **hourly**, specify:
 - **Start date** – select the date and time the automated upload should begin. Use the calendar picker and 24-hour time selector.
 - **End date** – select the date and time the automated upload should end.
 - **Days of the week** – select each day the automated upload schedule should run.
 - Select **custom**, to specify the interval granularity in seconds, minutes, or hours, as well as other date and time parameters.
6. In the Task Properties dialog, select the **Options** tab, and set the number of inactive days for which to purge.

For Subscription Cleanup, Error History Cleanup, and Client Log Cleanup, set the number of inactive days for which to purge. For Synchronization Cache Cleanup, set the purge interval for EIS managed caches. For more information on EIS-managed caches, see *EIS Defined Cache Partitions* in *Mobile Data Models: Using Mobile Business Objects* and

Creating Cache Groups in SAP Mobile WorkSpace - Mobile Business Object Development.

7. Click **OK** to save the schedule properties and purge options.
8. Select **Enable**. Schedules run until you disable them, or until they expire.

See also

- *Creating and Enabling a New Domain* on page 101
- *Deleting a Domain* on page 102
- *Registering a Domain Administrator User* on page 103
- *Assigning Domain Administrators to a Domain* on page 104
- *Viewing Applications for a Domain* on page 105
- *Viewing Application Connections for a Domain* on page 105
- *Domain Logs* on page 110
- *Client Logs* on page 162
- *Connections* on page 167
- *Configuring Domain Security* on page 195

Enabling Domain Cleanup

The SUP Administrator or SUP Domain Administrator must enable the schedule as a separate task.

You can set up the schedule, and enable it at a later time. Once enabled, the cleanup runs automatically until is changed, disabled, or expires. You can check the current enabled or disabled status on the **Scheduled Tasks** tab.

1. In the left navigation pane of SAP Control Center, select the cluster, expand the Domain folders folder, and select a domain.
2. In the right administration pane, select the **Scheduled Tasks** tab.
3. Select one of the cleanup options, and verify the value in the Status column is set to **disabled**.
4. On the **Scheduled Tasks** tab, click **Enable**.
5. Click **Yes** to confirm. The value in the Status column changes to **enabled**. The cleanup schedule runs automatically for the selected option.

Disabling Domain Cleanup

The SUP Administrator or SUP Domain Administrator can disable, or reenable, a scheduled cleanup option at any time.

If you disable the cleanup option while it is running, the current process continues. Future action is disabled, unless you reenable the option.

1. In the left navigation pane of SAP Control Center, select the cluster, expand the Domains folder, and select a domain.
2. In the right administration pane, select the **Scheduled Tasks** tab.
3. Select one of the cleanup options, and verify the value in the Status column is set to **enabled**.
4. On the **Scheduled** tab, click **Disable**.
5. Click **Yes** to confirm. The value in the Status column changes to **disabled**.

Running Manual Cleanup by Domain

At any time the SUP Administrator or SUP Domain Administrator can manually run cleanup options. The processes run asynchronously on SAP Mobile Server using the current settings.

As much as is reasonable, use manual cleanup when system load is light.

1. In the left pane, expand the **Domains** folder, and select a domain.
2. In the right pane, select the **Scheduled Tasks** tab.
3. Select one of the cleanup options.
4. Click **Run Now**, then optionally specify the purge options.

For Subscription Cleanup, Error History Cleanup, and Client Log Cleanup, set the number of inactive days for which to purge. For Synchronization Cache Cleanup, set the purge interval for EIS managed caches. For more information on EIS-managed caches, see *EIS Defined Cache Partitions* in *Mobile Data Models: Using Mobile Business Objects* and *Creating Cache Groups* in *SAP Mobile WorkSpace - Mobile Business Object Development*.
5. Click **OK (Yes)** to confirm. The request is sent immediately, and the task runs asynchronously on SAP Mobile Server.

Domain Logs

The domain log enables an administrator to monitor application activities throughout the system. Detailed views of application activities are available by subsystem. The administrator can review activities in a specific subsystem log view, view correlated data in multiple subsystems, or view a unified log across all subsystems. The administrator must enable logging, and then use log filters to view data of interest.

By default, only error messages are recorded in each domain's log. To enable domain logging, you must create a log profile. See *Creating and Enabling Domain Logging* in *SAP Control Center for SAP Mobile Platform*.

See also

- *Creating and Enabling a New Domain* on page 101
- *Deleting a Domain* on page 102
- *Registering a Domain Administrator User* on page 103
- *Assigning Domain Administrators to a Domain* on page 104

- *Viewing Applications for a Domain* on page 105
- *Viewing Application Connections for a Domain* on page 105
- *Scheduling Accumulated Data Cleanup for Domains* on page 107
- *Client Logs* on page 162
- *Connections* on page 167
- *Configuring Domain Security* on page 195

Creating and Enabling Domain Logging

Create logging profile definitions and enable the log profile.

1. Display the **Settings** tab for Domain Logs.
In the navigation pane, click **Domains** > <domainName> > **Log**, then select **Settings** from the administration pane.
2. Click **New**.
3. In the Profile Definition dialog, enter a **Name** and **Description** for the log profile.
4. Add the necessary profile definitions.
5. Select **Enable after creation**.
6. Click **OK**.

Note: To ensure the domain logs are populated immediately after enabling the log profile, do the following:

- a. Display the **Settings** tab for the default domain.
In the navigation pane, expand the **Domains** folder and select the default domain.
 - b. Select **Log**.
 - c. In the right administration pane, select the **Settings** tab.
 - d. Click **Configuration**.
 - e. Uncheck **Enable flush threshold**.
-

1. *Creating the Profile Definition*

Create profile definitions belonging to multiple categories.

2. *Enabling the Created Profile*

Enable the profile you have created to monitor the log profile definitions.

3. *Enabling and Configuring Domain Logging*

Configure auto purge, flush threshold, and flush batch size settings to determine how long domain log data is retained, how frequently it is written to database from server nodes, and set a domain log database connection to configure where domain log data is stored.

Creating the Profile Definition

Create profile definitions belonging to multiple categories.

You can add profile definitions by selecting applications, security configurations, users, connections, applications connections or payloads of your choice.

Note: To enable DCN request logging, you must include the following in the profile definition:

- for general DCN logging, an MBO package, MBO, user, or security configuration
- for Hybrid App DCN logging, an MBO package, user, or security configuration

To include payload data in the log, add payloads to the profile definition and select the appropriate subsystem.

See also

- *Enabling the Created Profile* on page 117

Adding or Removing Applications to the Profile

Add applications to the log profile that are currently deployed in the selected domain.

Note: If a logging profile includes both package related and payloads criteria, you may experience a serious performance impact. Make sure you include additional criteria such as users or application connections.

1. In the Profile Definition dialog, select **Package related**.
2. Select **Applications**, then click the button to add application to the profile.
The Applications dialog is displayed with the list of applications currently deployed in this domain.
3. To search for the application you want to add to the profile, select the search criteria from the **Search** drop-down list and enter a value for this criteria.
4. Click **Go**.
5. You can do any of the following:
 - To add an application to a new or existing profile, select the check-box adjacent to the application entry in the list.
 - To remove an application from the profile, uncheck the check-box adjacent to the application entry in the list.
6. Click **OK**.

See also

- *Adding or Removing Packages to the Profile* on page 113
- *Adding or Removing MBOs to the Profile* on page 113
- *Adding or Removing Operations to the Profile* on page 114

- *Adding or Removing Security Configurations to the Profile* on page 115
- *Adding or Removing Package Users to the Profile* on page 115
- *Adding or Removing Connections to the Profile* on page 116
- *Adding or Removing Application Connections to the Profile* on page 116
- *Adding or Removing Payloads to the Profile* on page 117

Adding or Removing Packages to the Profile

Add one or more packages to the profile.

Note: If a logging profile includes both package related and payloads criteria, you may experience a serious performance impact. Make sure you include additional criteria such as users or application connections.

1. In the Profile Definition dialog, select **Package related**.
2. Select **Packages**, then click the button to add packages to the profile.
3. Select one or more packages to include.
4. Click **OK**.

Note: To remove a selection, click **View selection**, select the item, then click **Remove**.

See also

- *Adding or Removing Applications to the Profile* on page 112
- *Adding or Removing MBOs to the Profile* on page 113
- *Adding or Removing Operations to the Profile* on page 114
- *Adding or Removing Security Configurations to the Profile* on page 115
- *Adding or Removing Package Users to the Profile* on page 115
- *Adding or Removing Connections to the Profile* on page 116
- *Adding or Removing Application Connections to the Profile* on page 116
- *Adding or Removing Payloads to the Profile* on page 117

Adding or Removing MBOs to the Profile

Add MBOs from one or more packages to the profile.

Note: If a logging profile includes both package related and payloads criteria, you may experience a serious performance impact. Make sure you include additional criteria such as users or application connections.

1. In the Profile Definition dialog, select **Package related**.
2. Select **MBOs**, then click the button to add MBOs to the profile.
3. Navigate to a package in the left pane.
4. Select the MBO in the right pane.

5. Identify another MBO, or click **OK**.

Note: To remove a selection, click **View selection**, select the item, then click **Remove**.

See also

- *Adding or Removing Applications to the Profile* on page 112
- *Adding or Removing Packages to the Profile* on page 113
- *Adding or Removing Operations to the Profile* on page 114
- *Adding or Removing Security Configurations to the Profile* on page 115
- *Adding or Removing Package Users to the Profile* on page 115
- *Adding or Removing Connections to the Profile* on page 116
- *Adding or Removing Application Connections to the Profile* on page 116
- *Adding or Removing Payloads to the Profile* on page 117

Adding or Removing Operations to the Profile

Add operations to the log profile for MBOs that are currently deployed in the selected domain. You can select operations for one or more MBOs.

Note: If a logging profile includes both package related and payloads criteria, you may experience a serious performance impact. Make sure you include additional criteria such as users or application connections.

1. In the Profile Definition dialog, select **Package related**.
2. Select **Operations**, then click the button to add operations to the profile.
3. Navigate to a package in the left pane, and select an MBO.
4. Identify the create, delete, or update operations to include for the MBO in the right pane.
5. Select another MBO and its operations, or click **OK**.

Note: To remove a selection, click **View selection**, select the item, then click **Remove**.

See also

- *Adding or Removing Applications to the Profile* on page 112
- *Adding or Removing Packages to the Profile* on page 113
- *Adding or Removing MBOs to the Profile* on page 113
- *Adding or Removing Security Configurations to the Profile* on page 115
- *Adding or Removing Package Users to the Profile* on page 115
- *Adding or Removing Connections to the Profile* on page 116
- *Adding or Removing Application Connections to the Profile* on page 116
- *Adding or Removing Payloads to the Profile* on page 117

Adding or Removing Security Configurations to the Profile

Add one or more domain security configurations to the profile.

1. In the Profile Definition dialog, select **Security related**.
2. Select **Security configuration**, then click the button to add security configurations to the profile.
3. Select one or more security configuration to include.
4. Click **OK**.

Note: To remove a selection, click **View selection**, select the item, then click **Remove**.

See also

- *Adding or Removing Applications to the Profile* on page 112
- *Adding or Removing Packages to the Profile* on page 113
- *Adding or Removing MBOs to the Profile* on page 113
- *Adding or Removing Operations to the Profile* on page 114
- *Adding or Removing Package Users to the Profile* on page 115
- *Adding or Removing Connections to the Profile* on page 116
- *Adding or Removing Application Connections to the Profile* on page 116
- *Adding or Removing Payloads to the Profile* on page 117

Adding or Removing Package Users to the Profile

Add one or more domain package users to the profile to include security configurations.

1. In the Profile Definition dialog, select **Security related**.
2. Select **Users**, then click the button to add package users to the profile.
3. Select one or more user name to include.
4. Click **OK**.

Note: To remove a selection, click **View selection**, select the item, then click **Remove**.

See also

- *Adding or Removing Applications to the Profile* on page 112
- *Adding or Removing Packages to the Profile* on page 113
- *Adding or Removing MBOs to the Profile* on page 113
- *Adding or Removing Operations to the Profile* on page 114
- *Adding or Removing Security Configurations to the Profile* on page 115
- *Adding or Removing Connections to the Profile* on page 116
- *Adding or Removing Application Connections to the Profile* on page 116

- *Adding or Removing Payloads to the Profile* on page 117

Adding or Removing Connections to the Profile

Add connections of a particular connection type to the profile.

1. In the Profile Definition dialog, select **Connections**.
2. Click the button to add connections to the profile.
The Connections dialog is displayed with the list of connections currently created in this domain.
3. You can do any of the following:
 - To add a connection to a new or existing profile, select a **Connection Type** entry in the list and select the connection names.
 - To remove an application from the profile, select the **View selection** check-box. Select the connection type and click **Remove**.
4. Click **OK**.

See also

- *Adding or Removing Applications to the Profile* on page 112
- *Adding or Removing Packages to the Profile* on page 113
- *Adding or Removing MBOs to the Profile* on page 113
- *Adding or Removing Operations to the Profile* on page 114
- *Adding or Removing Security Configurations to the Profile* on page 115
- *Adding or Removing Package Users to the Profile* on page 115
- *Adding or Removing Application Connections to the Profile* on page 116
- *Adding or Removing Payloads to the Profile* on page 117

Adding or Removing Application Connections to the Profile

Add application connections to the profile. This enables you to list current application connections to SAP Mobile Server.

1. In Profile Definition, select **Application connections**.
2. Click the button to add application connections to the profile.
3. In Application Connections, select Application Connections, and the data columns to include
4. Click **OK**.
5. To remove an application connection from the profile, select the **View selection** check-box. Select the application connection and click **Remove**.

See also

- *Adding or Removing Applications to the Profile* on page 112

- *Adding or Removing Packages to the Profile* on page 113
- *Adding or Removing MBOs to the Profile* on page 113
- *Adding or Removing Operations to the Profile* on page 114
- *Adding or Removing Security Configurations to the Profile* on page 115
- *Adding or Removing Package Users to the Profile* on page 115
- *Adding or Removing Connections to the Profile* on page 116
- *Adding or Removing Payloads to the Profile* on page 117

Adding or Removing Payloads to the Profile

Add a subsystem to the profile for payload logging. This enables you to identify one or more specific subsystems where payload data will also be included with the logged activity. If specified, payload is enabled for the selected subsystems.

The payload corresponds to the information of one request serviced by the SAP Mobile Server. Example: For an administrator to keep track of granular details such as request headers sent to Gateway through the proxy server, payload is enabled for a profile.

Note: If a logging profile includes both package related and payloads criteria, you may experience a serious performance impact. Make sure you include additional criteria such as users or application connections.

1. In the Profile Definition dialog, select **Payloads**.
2. Click the button to add payloads to the profile.
The Payloads dialog is displayed.
3. You can do any of the following:
 - Select the check-box adjacent to the subsystem you want to add to the profile.
 - Uncheck the check-box that you want to remove from the profile.
4. Click **OK**.

See also

- *Adding or Removing Applications to the Profile* on page 112
- *Adding or Removing Packages to the Profile* on page 113
- *Adding or Removing MBOs to the Profile* on page 113
- *Adding or Removing Operations to the Profile* on page 114
- *Adding or Removing Security Configurations to the Profile* on page 115
- *Adding or Removing Package Users to the Profile* on page 115
- *Adding or Removing Connections to the Profile* on page 116
- *Adding or Removing Application Connections to the Profile* on page 116

Enabling the Created Profile

Enable the profile you have created to monitor the log profile definitions.

1. In the Profile Definition dialog, select **Enable after creation** to enable the logging profile once you have created it.
2. Click **OK**.

You can alternatively enable the log profile by doing the following:

- a. In the **Settings** tab, select the log profile you have created.
- b. Click **Enable**.
- c. Click **Yes** on the confirmation dialog.

See also

- *Creating the Profile Definition* on page 112

Enabling and Configuring Domain Logging

Configure auto purge, flush threshold, and flush batch size settings to determine how long domain log data is retained, how frequently it is written to database from server nodes, and set a domain log database connection to configure where domain log data is stored.

If you do not configure the auto-purge schedule, you can purge data manually with the **Purge** button. If you are manually purging logs with hundreds of thousands of entries, note that SAP Mobile Server removes these entries asynchronously to avoid negatively impacting runtime performance. For smaller logs, the purge action tends to be more instantaneous. To avoid large logs, use the auto purge schedule.

1. In the left navigation pane of SAP Control Center, select **Domains**.
2. Under the default domain node, select **Log**.
3. In the right administration pane, select the **Settings** tab. These settings are used for all domains.
4. Click **Configuration**.
5. Configure auto purge settings.

Auto purge clears obsolete data from the database once it reaches the specified threshold.

- a) Select **Enable auto-purge configuration** to activate auto purge functionality.
- b) Enter the length of time (in days) to retain monitoring data before it is purged.
- c) Specify the Purge time. Auto-purge executes once per day at the specified time.

6. Configure flush threshold settings:

The flush threshold indicates how often data is flushed from memory to the database. This allows you to specify the size of the data saved in memory before it is cleared. Alternately, if you do not enable a flush threshold, data is immediately written to the domain log database as it is captured.

- a) Select **Enable flush threshold** to activate flush threshold functionality.

Note: Enabling flush configuration is a good practice for performance considerations. Be aware there may be a consequent delay in viewing data, until data is stored in the database.

b) Select one of:

- **Number of rows** – domain log data that surpasses the specified number of rows is flushed from memory. Enter the desired number of rows adjacent to **Rows**. Disabled by default.
- **Time interval** – domain log data older than the specified time interval is flushed from memory. Enter the desired duration adjacent to **Minutes**. The default is 5.
- **Either rows or time interval** – domain log data is flushed from memory according to whichever value is reached first: either the specified number of rows or the specified time interval. Enter the desired rows and duration adjacent to **Rows** and **Minutes**, respectively.

7. If you enabled a flush threshold, enter a **Flush batch row size** by specifying the size of each batch of data sent to the domain log database. The row size must be a positive integer. The batch size divides flushed data into smaller segments, rather than saving all data together according to the flush threshold parameters. For example, if you set the flush threshold to 100 rows and the flush batch row size to 50, once 100 rows are collected in the console, the save process executes twice; data is flushed into the database in two batches of 50 rows. If the flush threshold is not enabled, the flush batch row size is implicitly 1.

Note: By default, the domain log database flushes data every 5 minutes. Alternatively, you can flush data immediately by removing or decreasing the default values, but doing so impacts performance.

8. Optional. To change the data source, select an available database from the **Domain log database endpoint** drop down list.

Available databases are those with a JDBC server connection type (SQL Anywhere) created in the default domain. To create a new database, a platform administrator must set up a database by running the appropriate configuration scripts and creating a server connection for the database in the default domain. The database server connection then appears as an option in the Domain Log Database Endpoint drop down list.

9. Optional. Change the maximum length of the payload data logged in the payload column(s) of each sub-system. Large payload content is truncated to the length specified as that value. The default max size is 12K (in bytes) which is configured in the 'default' domain and applicable for all domains. Increasing the domain payload size should be tested to identify proper configuration for the server's JVM memory settings.
10. Click **OK**.

Reviewing Domain Log Data

An administrator reviews logged data by creating log filters. The filters enable you to retrieve data logged for a specific thread, application, user, connection, among other options.

You can retrieve log data without using any filters, however, when there is large number of activities being logged, it may be advisable to filter the results to a more manageable size by specifying search conditions in the log filter (user, application, or thread-id).

You can combine multiple log filters that are common with sub-system specific filters when viewing in a sub-system view, and combine multiple sub-system filters in the ALL tab to retrieve the data of interest.

Supported Log Subsystems

Log subsystems provide categories that enable you to filter and correlate application data at a more granular level. Understanding these subsystems enables you to formulate more specific filters for tracking application activities throughout the system.

Subsystem	Description
All	Provides a unified view of all subsystems, enabling you to look for activities, trends, and symptoms across multiple subsystems.
Synchronization	Provides a view of synchronization activities. Within this subsystem, additional categories include data synchronization, operation replay, subscription, result checker, cache refresh, and data services (DS) interface.
Device Notification	Provides a view of device notification activities.
DCN	Provides a view of data change notification (DCN) activities. Within this subsystem, additional categories include general DCN, and Hybrid App DCN.
Security	Provides a view of security-related activities.
Error	Provides a view of errors.
Connection	Provides a view of connection activities. Within this subsystem, additional categories include DOE, JDBC, REST, SAP®, and SOAP connections.
Push	Provides a few of push notification activities.
Proxy	Provides a view of Online Data Proxy connection-related activities.
Server	Provides a view of server-related activities.
Dispatcher	Provides a view of dispatcher activities. Within this subsystem, categories include Replication, Messaging, and Service.

Subsystem	Description
Application	Provides a view of application activities. Within this subsystem, categories include Registration, Setting, Agency Event and Agency Message.

See also

- *Setting Up a Pool of Log Filters* on page 121
- *Retrieving Unified View Logs* on page 124
- *Retrieving Synchronization Logs* on page 125
- *Retrieving Device Notification Logs* on page 126
- *Retrieving Data Change Notification Logs* on page 127
- *Retrieving Security Logs* on page 128
- *Retrieving Error Logs* on page 129
- *Retrieving Connection Logs* on page 130
- *Retrieving Push Logs* on page 131
- *Retrieving Proxy Logs* on page 132
- *Retrieving Server Logs* on page 133
- *Retrieving Dispatcher Logs* on page 134
- *Retrieving Application Logs* on page 135
- *Correlating Log Data Across Subsystems* on page 136
- *Exporting Log Data* on page 137

Setting Up a Pool of Log Filters

Set up a pool of log filters to filter out unwanted application activities, and provide a view of specific activities. Use SAP Control Center to create and manage your filters.

See also

- *Supported Log Subsystems* on page 120
- *Retrieving Unified View Logs* on page 124
- *Retrieving Synchronization Logs* on page 125
- *Retrieving Device Notification Logs* on page 126
- *Retrieving Data Change Notification Logs* on page 127
- *Retrieving Security Logs* on page 128
- *Retrieving Error Logs* on page 129
- *Retrieving Connection Logs* on page 130
- *Retrieving Push Logs* on page 131
- *Retrieving Proxy Logs* on page 132
- *Retrieving Server Logs* on page 133

- *Retrieving Dispatcher Logs* on page 134
- *Retrieving Application Logs* on page 135
- *Correlating Log Data Across Subsystems* on page 136
- *Exporting Log Data* on page 137

Reusable Log Filters

Create reusable log filters that you can use as a base. One strategy is to create a base log filter for each of the supported log subsystems, and for significant categories within subsystems. Another strategy is to create common log filters (useful across subsystems) on specific criteria, such as thread ID, user, package, and so forth.

You can modify these base log filters as needed for more specific searches, or clone the log filter and modify it for a specific search.

See also

- *Creating Log Filters* on page 122
- *Deleting Filters* on page 123
- *Updating Filters* on page 124

Creating Log Filters

Filter the log data by creating filters across subsystems that define the appropriate search criteria.

1. In the left navigation pane of the SAP Control Center, select the **Domains** node.
2. Select the domain node and select the **Log** node.
3. In the right administration pane, select the **General** tab.
4. Click a subsystem and check **Show filter**.
5. Select + to add a filter definition to a subsystem.
6. In the Filter Definition dialog, enter the **Name** and **Description** of the filter.
7. Select the **Sub System**.
8. Select the filter criteria and assign values to the criteria selected. You can use the logical operations to compose the criteria.

Note: You use the 'AND' logical operator to highlight filter relations belonging to the same subsystem. Filter definitions among multiple subsystems use the 'OR' logical operator.

9. Click **OK**.

See also

- *Reusable Log Filters* on page 122
- *Deleting Filters* on page 123
- *Updating Filters* on page 124

Domain Log Filter Guidelines

Domain log filters are divided into three groups: package related, security related and application connections. There are important guidelines you should consider when you are creating domain log filters, since certain combinations may not work as expected.

- The "OR" logical operator is used for the filters in same group. The domain log is generated only if there is at least one filter group allowing the subsystem, and no filter group disallowing it.
- A filter is applied only if the subsystem includes the corresponding column, and the column has a value when logging. For example, the operation filter in the package-related group can be applied for Application and Operation Replay; Application and DS Interface; and all Connection subsystems. If the operation has no value in the logging context when logging (which is possible for Application and DS Interface, and all Connection subsystems), the operation filter does not take effect.
- If one filter is not applied, the permission is neither allowed nor disallowed, and other filters (groups) determine whether to generate domain logs. If no filter group is allowed, the domain log is not generated.

For example, say a filter is set for both operation and security:

- For subsystems that do not apply the operation filter—such as Data Sync, Cache Refresh and Device Notification—domain logs are generated only for security configuration filter matches.
- For the Operation Replay subsystem, domain logs are generated only for operations filter matches.
- For the DS Interface and Connection subsystems, domain logs are generated only for operation filter matches and null generated domain logs.

If only the operation filter is selected, then only operations matching the operation filter generate domain logs in the Operation Replay, DS Interface, and Connection subsystems. Subsystems that do not apply operation filters do not generate any domain logs.

Deleting Filters

Delete the filters created for sub systems

1. Display the **General** tab for Domain Logs.

In the navigation pane, click **Domain** > <domainName> > **Log**, then select **General** from the administration pane.

2. Select **Show filter**.
3. Select the filter from the list.
4. Click **X** to delete the filter.

See also

- *Reusable Log Filters* on page 122
- *Creating Log Filters* on page 122
- *Updating Filters* on page 124

Updating Filters

Update filters as needed to fine tune log file filtering.

1. Display the **General** tab for Domain Logs.

In the navigation pane, click **Domain** > <domainName> > **Log**, then select **General** from the administration pane.

2. Select **Show filter**.
3. Select the filter from the list.
4. Click the properties icon to review or modify the filter.

Note: Alternatively, click the clone icon to clone the filter, then proceed to modify it.

5. In Filter Definition, modify the description, and set up the filter criteria.
6. Click **OK**.

See also

- *Reusable Log Filters* on page 122
- *Creating Log Filters* on page 122
- *Deleting Filters* on page 123

Retrieving Unified View Logs

Retrieves logging data across the domain to provide a unified view.

1. Display the **General** tab for Domain Logs.

In the navigation pane, click **Domains** > <domainName> > **Log**, then select **General** from the administration pane.

2. Select the **All** tab.
3. To filter the display, select **Show filter** and either:
 - Use an existing filter by checking the box adjacent to the filter name.
 - Create a new filter by clicking + and choosing a starting date and time, and ending date and time.

To export the filtered results, click **Export** and select an appropriate output destination.

4. Click **Retrieve** to retrieve the logs.
The table is populated with the list of logs.

See also

- *Supported Log Subsystems* on page 120
- *Setting Up a Pool of Log Filters* on page 121
- *Retrieving Synchronization Logs* on page 125
- *Retrieving Device Notification Logs* on page 126
- *Retrieving Data Change Notification Logs* on page 127
- *Retrieving Security Logs* on page 128
- *Retrieving Error Logs* on page 129
- *Retrieving Connection Logs* on page 130
- *Retrieving Push Logs* on page 131
- *Retrieving Proxy Logs* on page 132
- *Retrieving Server Logs* on page 133
- *Retrieving Dispatcher Logs* on page 134
- *Retrieving Application Logs* on page 135
- *Correlating Log Data Across Subsystems* on page 136
- *Exporting Log Data* on page 137

Retrieving Synchronization Logs

Retrieve logging data related to different aspects of data synchronization, including data, subscriptions, operations, result checker, cache refresh and the data service and SAP Mobile Server interface.

With data in these logs and the correlation tool, follow the data path between the enterprise information system (EIS), SAP Mobile Server, cache database, and user application connection.

1. Display the **General tab for Domain Logs.**

In the navigation pane, click **Domains** > **<domainName>** > **Log**, then select **General** from the administration pane.

2. Select the **Synchronization tab.****3. Select a synchronization subsystem:**

- Data Sync – view data synchronization details.
- Operation Replay – view MBO operational replay details.
- Subscription – view subscription details.
- Result Checker – view result checker details.
- Cache Refresh – view cache database interaction details.
- DS Interface – view requests entering data services that result in interaction with cache database and EIS.

4. To filter the display, select **Show filter and either:**

- Use an existing filter by checking the box adjacent to the filter name.

- Create a new filter by clicking + and choosing a starting date and time, and ending date and time.

To customize the display, select the view type (grid or text). To export the filtered results, click **Export** and select an appropriate output destination.

5. (Optional) Select the columns to display from the drop down list, such as Application ID, Application Connection Id, and User.
6. Click **Retrieve** to retrieve the logs.
7. (Optional) Select a specific row to view additional columns in the detail area.

See also

- *Supported Log Subsystems* on page 120
- *Setting Up a Pool of Log Filters* on page 121
- *Retrieving Unified View Logs* on page 124
- *Retrieving Device Notification Logs* on page 126
- *Retrieving Data Change Notification Logs* on page 127
- *Retrieving Security Logs* on page 128
- *Retrieving Error Logs* on page 129
- *Retrieving Connection Logs* on page 130
- *Retrieving Push Logs* on page 131
- *Retrieving Proxy Logs* on page 132
- *Retrieving Server Logs* on page 133
- *Retrieving Dispatcher Logs* on page 134
- *Retrieving Application Logs* on page 135
- *Correlating Log Data Across Subsystems* on page 136
- *Exporting Log Data* on page 137

Retrieving Device Notification Logs

Retrieves logging data for server-initiated synchronization notifications between SAP Mobile Server and devices.

1. Display the **General** tab for Domain Logs.

In the navigation pane, click **Domain** > *<domainName>* > **Log**, then select **General** from the administration pane.

2. Select the **Device Notification** tab.

3. To filter the display, select **Show filter** and either:

- Use an existing filter by checking the box adjacent to the filter name.
- Create a new filter by clicking + and choosing a starting date and time, and ending date and time.

To customize the display, select the view type (grid or text), or click >> to show only selected columns. To export the filtered results, click **Export** and select an appropriate output destination.

4. (Optional) Select the columns to display from the drop down list, such as Application ID, Application Connection Id, and User.
5. Click **Retrieve** to retrieve the log data.
6. (Optional) Select a specific row to view additional columns in the detail area.

See also

- *Supported Log Subsystems* on page 120
- *Setting Up a Pool of Log Filters* on page 121
- *Retrieving Unified View Logs* on page 124
- *Retrieving Synchronization Logs* on page 125
- *Retrieving Data Change Notification Logs* on page 127
- *Retrieving Security Logs* on page 128
- *Retrieving Error Logs* on page 129
- *Retrieving Connection Logs* on page 130
- *Retrieving Push Logs* on page 131
- *Retrieving Proxy Logs* on page 132
- *Retrieving Server Logs* on page 133
- *Retrieving Dispatcher Logs* on page 134
- *Retrieving Application Logs* on page 135
- *Correlating Log Data Across Subsystems* on page 136
- *Exporting Log Data* on page 137

Retrieving Data Change Notification Logs

Retrieves logging data for data change notifications (DCN) between an enterprise information system (EIS) and an MBO package, for general and Hybrid App DCN.

1. Display the **General** tab for Domain Logs.
In the navigation pane, click **Domain** > <domainName> > **Log**, then select **General** from the administration pane.
2. Select the **DCN** tab.
3. Select the DCN type:
 - General DCN
 - Hybrid App DCN
4. To filter the display, select **Show filter** and either:
 - Use an existing filter by checking the box adjacent to the filter name.

- Create a new filter by clicking + and choosing a starting date and time, and ending date and time.

To customize the display, select the view type (grid or text), or click >> to show only selected columns. To export the filtered results, click **Export** and select an appropriate output destination.

5. (Optional) Select the columns to display from the drop down list, such as Application ID, Application Connection Id, and User.
6. Click **Retrieve** to retrieve the logs.
7. (Optional) Select a specific row to view additional columns in the detail area.

See also

- *Supported Log Subsystems* on page 120
- *Setting Up a Pool of Log Filters* on page 121
- *Retrieving Unified View Logs* on page 124
- *Retrieving Synchronization Logs* on page 125
- *Retrieving Device Notification Logs* on page 126
- *Retrieving Security Logs* on page 128
- *Retrieving Error Logs* on page 129
- *Retrieving Connection Logs* on page 130
- *Retrieving Push Logs* on page 131
- *Retrieving Proxy Logs* on page 132
- *Retrieving Server Logs* on page 133
- *Retrieving Dispatcher Logs* on page 134
- *Retrieving Application Logs* on page 135
- *Correlating Log Data Across Subsystems* on page 136
- *Exporting Log Data* on page 137

Retrieving Security Logs

Retrieves security details for specific applications.

1. Display the **General** tab for Domain Logs.
In the navigation pane, click **Domain** > <domainName> > **Log**, then select **General** from the administration pane.
2. Select the **Security** tab.
3. To filter the display, select **Show filter** and either:
 - Use an existing filter by checking the box adjacent to the filter name.
 - Create a new filter by clicking + and choosing a starting date and time, and ending date and time.

To customize the display, select the view type (grid or text), or click >> to show only selected columns. To export the filtered results, click **Export** and select an appropriate output destination.

4. (Optional) Select the columns to display from the drop down list, such as Application ID, Application Connection Id, and User.
5. Click **Retrieve** to retrieve the logs.

See also

- *Supported Log Subsystems* on page 120
- *Setting Up a Pool of Log Filters* on page 121
- *Retrieving Unified View Logs* on page 124
- *Retrieving Synchronization Logs* on page 125
- *Retrieving Device Notification Logs* on page 126
- *Retrieving Data Change Notification Logs* on page 127
- *Retrieving Error Logs* on page 129
- *Retrieving Connection Logs* on page 130
- *Retrieving Push Logs* on page 131
- *Retrieving Proxy Logs* on page 132
- *Retrieving Server Logs* on page 133
- *Retrieving Dispatcher Logs* on page 134
- *Retrieving Application Logs* on page 135
- *Correlating Log Data Across Subsystems* on page 136
- *Exporting Log Data* on page 137

Retrieving Error Logs

Retrieves logging data for domain errors. Note that error logging is always on, and any error that occurs for any application activity is logged.

1. Display the **General** tab for Domain Logs.

In the navigation pane, click **Domain** > *<domainName>* > **Log**, then select **General** from the administration pane.

2. Select the **Error** tab.
3. To filter the display, select **Show filter** and either:
 - Use an existing filter by checking the box adjacent to the filter name.
 - Create a new filter by clicking + and choosing a starting date and time, and ending date and time.

To customize the display, select the view type (grid or text), or click >> to show only selected columns. To export the filtered results, click **Export** and select an appropriate output destination.

4. (Optional) Select the columns to display from the drop down list, such as Application ID, Application Connection Id, and User.
5. Click **Retrieve** to retrieve the logs.

See also

- *Supported Log Subsystems* on page 120
- *Setting Up a Pool of Log Filters* on page 121
- *Retrieving Unified View Logs* on page 124
- *Retrieving Synchronization Logs* on page 125
- *Retrieving Device Notification Logs* on page 126
- *Retrieving Data Change Notification Logs* on page 127
- *Retrieving Security Logs* on page 128
- *Retrieving Connection Logs* on page 130
- *Retrieving Push Logs* on page 131
- *Retrieving Proxy Logs* on page 132
- *Retrieving Server Logs* on page 133
- *Retrieving Dispatcher Logs* on page 134
- *Retrieving Application Logs* on page 135
- *Correlating Log Data Across Subsystems* on page 136
- *Exporting Log Data* on page 137

Retrieving Connection Logs

Retrieves logging data for domain connections for specific connection types to backend data sources, including DOE, JDBC, REST, SAP, and SOAP if enabled.

1. Display the **General** tab for Domain Logs.
In the navigation pane, click **Domain** > <domainName> > **Log**, then select **General** from the administration pane.
2. Select the **Connection** tab.
3. Select a connection type, such as DOE, JDBC, REST, SAP, or SOAP.
4. To filter the display, select **Show filter** and either:
 - Use an existing filter by checking the box adjacent to the filter name.
 - Create a new filter by clicking + and choosing a starting date and time, and ending date and time.

To customize the display, select the view type (grid or text), or click >> to show only selected columns. To export the filtered results, click **Export** and select an appropriate output destination.

5. (Optional) Select the columns to display from the drop down list, such as Application ID, Application Connection Id, and User.

6. Click **Retrieve** to retrieve the data.
7. (Optional) Select a specific row to view additional columns in the detail area.

See also

- *Supported Log Subsystems* on page 120
- *Setting Up a Pool of Log Filters* on page 121
- *Retrieving Unified View Logs* on page 124
- *Retrieving Synchronization Logs* on page 125
- *Retrieving Device Notification Logs* on page 126
- *Retrieving Data Change Notification Logs* on page 127
- *Retrieving Security Logs* on page 128
- *Retrieving Error Logs* on page 129
- *Retrieving Push Logs* on page 131
- *Retrieving Proxy Logs* on page 132
- *Retrieving Server Logs* on page 133
- *Retrieving Dispatcher Logs* on page 134
- *Retrieving Application Logs* on page 135
- *Correlating Log Data Across Subsystems* on page 136
- *Exporting Log Data* on page 137

Retrieving Push Logs

Retrieves the log data for all push notifications.

Once you have traced a connection under the Applications node, you can retrieve the logs under the Domains node.

1. Display the **General** tab for Domain Logs.

In the navigation pane, click **Domain** > <domainName> > **Log**, then select **General** from the administration pane.

2. Select **Push**.

3. To filter the display, select **Show filter** and either:

- Use an existing filter by checking the box adjacent to the filter name.
- Create a new filter by clicking + and choosing a starting date and time, and ending date and time.

To customize the display, select the view type (grid or text), or click >> to show only selected columns. To export the filtered results, click **Export** and select an appropriate output destination.

4. (Optional) Select the columns to display from the drop down list, such as Application ID, Application Connection Id, and User.

5. Click **Retrieve** to retrieve the data.
6. (Optional) Select a specific row to view additional columns in the detail area.

See also

- *Supported Log Subsystems* on page 120
- *Setting Up a Pool of Log Filters* on page 121
- *Retrieving Unified View Logs* on page 124
- *Retrieving Synchronization Logs* on page 125
- *Retrieving Device Notification Logs* on page 126
- *Retrieving Data Change Notification Logs* on page 127
- *Retrieving Security Logs* on page 128
- *Retrieving Error Logs* on page 129
- *Retrieving Connection Logs* on page 130
- *Retrieving Proxy Logs* on page 132
- *Retrieving Server Logs* on page 133
- *Retrieving Dispatcher Logs* on page 134
- *Retrieving Application Logs* on page 135
- *Correlating Log Data Across Subsystems* on page 136
- *Exporting Log Data* on page 137

Retrieving Proxy Logs

Retrieves the log data for all requests and responses made from the Proxy server.

Once you have traced a connection under the Applications node, you can retrieve the logs under the Domains node.

1. Display the **General** tab for Domain Logs.

In the navigation pane, click **Domain** > <domainName> > **Log**, then select **General** from the administration pane.

2. Select the **Proxy**.

3. To filter the display, select **Show filter** and either:

- Use an existing filter by checking the box adjacent to the filter name.
- Create a new filter by clicking + and choosing a starting date and time, and ending date and time.

To customize the display, select the view type (grid or text), or click >> to show only selected columns. To export the filtered results, click **Export** and select an appropriate output destination.

4. (Optional) Select the columns to display from the drop down list, such as Application ID, Application Connection Id, and User.

5. Click **Retrieve** to retrieve the log data.
6. (Optional) Select a specific row to view additional columns in the detail area.

See also

- *Supported Log Subsystems* on page 120
- *Setting Up a Pool of Log Filters* on page 121
- *Retrieving Unified View Logs* on page 124
- *Retrieving Synchronization Logs* on page 125
- *Retrieving Device Notification Logs* on page 126
- *Retrieving Data Change Notification Logs* on page 127
- *Retrieving Security Logs* on page 128
- *Retrieving Error Logs* on page 129
- *Retrieving Connection Logs* on page 130
- *Retrieving Push Logs* on page 131
- *Retrieving Server Logs* on page 133
- *Retrieving Dispatcher Logs* on page 134
- *Retrieving Application Logs* on page 135
- *Correlating Log Data Across Subsystems* on page 136
- *Exporting Log Data* on page 137

Retrieving Server Logs

Retrieves logging data for domain servers.

1. Display the **General** tab for Domain Logs.
In the navigation pane, click **Domain** > <domainName> > **Log**, then select **General** from the administration pane.
2. Select the **Server** tab.
3. To filter the display, select **Show filter** and either:
 - Use an existing filter by checking the box adjacent to the filter name.
 - Create a new filter by clicking + and choosing a starting date and time, and ending date and time.

To customize the display, select the view type (grid or text), or click >> to show only selected columns. To export the filtered results, click **Export** and select an appropriate output destination.
4. (Optional) Select the columns to display from the drop down list, such as Application ID, Application Connection Id, and User.
5. Click **Retrieve** to retrieve the logs.

See also

- *Supported Log Subsystems* on page 120
- *Setting Up a Pool of Log Filters* on page 121
- *Retrieving Unified View Logs* on page 124
- *Retrieving Synchronization Logs* on page 125
- *Retrieving Device Notification Logs* on page 126
- *Retrieving Data Change Notification Logs* on page 127
- *Retrieving Security Logs* on page 128
- *Retrieving Error Logs* on page 129
- *Retrieving Connection Logs* on page 130
- *Retrieving Push Logs* on page 131
- *Retrieving Proxy Logs* on page 132
- *Retrieving Dispatcher Logs* on page 134
- *Retrieving Application Logs* on page 135
- *Correlating Log Data Across Subsystems* on page 136
- *Exporting Log Data* on page 137

Retrieving Dispatcher Logs

Retrieves logging data related to different aspects of the dispatcher, including messaging, and service data. Using data in these logs and the correlation tool, you can follow the data path between the enterprise information system (EIS), SAP Mobile Server, cache database, and user application connection.

1. Display the **General** tab for Domain Logs.

In the navigation pane, click **Domain** > <domainName> > **Log**, then select **General** from the administration pane.

2. Select the **Dispatcher** tab.
3. Select a dispatcher subsystem: Messaging, or Service.
4. To filter the display, select **Show filter** and either:
 - Use an existing filter by checking the box adjacent to the filter name.
 - Create a new filter by clicking + and choosing a starting date and time, and ending date and time.

To customize the display, select the view type (grid or text), or click >> to show only selected columns. To export the filtered results, click **Export** and select an appropriate output destination.

5. (Optional) Select the columns to display from the drop down list, such as Application ID, Application Connection Id, and User.
6. Click **Retrieve** to retrieve the logs.
7. (Optional) Select a specific row to view additional columns in the detail area.

See also

- *Supported Log Subsystems* on page 120
- *Setting Up a Pool of Log Filters* on page 121
- *Retrieving Unified View Logs* on page 124
- *Retrieving Synchronization Logs* on page 125
- *Retrieving Device Notification Logs* on page 126
- *Retrieving Data Change Notification Logs* on page 127
- *Retrieving Security Logs* on page 128
- *Retrieving Error Logs* on page 129
- *Retrieving Connection Logs* on page 130
- *Retrieving Push Logs* on page 131
- *Retrieving Proxy Logs* on page 132
- *Retrieving Server Logs* on page 133
- *Retrieving Application Logs* on page 135
- *Correlating Log Data Across Subsystems* on page 136
- *Exporting Log Data* on page 137

Retrieving Application Logs

Retrieves application logging data, including registration and setting information, and Agency Server events and client-server messages for Agency applications. Using data in the registration and setting logs and the correlation tool, you can follow the data path between the enterprise information system (EIS), SAP Mobile Server, cache database, and user application connection.

1. Display the **General** tab for Domain Logs.

In the navigation pane, click **Domain** > *<domainName>* > **Log**, then select **General** from the administration pane.

2. Select the **Application** tab.
3. Select a application subsystem: Registration, Setting, Agency Event, or Agency Message.
4. To filter the display, select **Show filter** and either:
 - Use an existing filter by checking the box adjacent to the filter name.
 - Create a new filter by clicking + and choosing a starting date and time, and ending date and time.

To customize the display, select the view type (grid or text), or click >> to show only selected columns. To export the filtered results, click **Export** and select an appropriate output destination.

5. (Optional) Select the columns to display from the drop down list, such as Application ID, Application Connection Id, and User.

6. Click **Retrieve** to retrieve the logs.
7. (Optional) Select a specific row to view additional columns in the detail area.

See also

- *Supported Log Subsystems* on page 120
- *Setting Up a Pool of Log Filters* on page 121
- *Retrieving Unified View Logs* on page 124
- *Retrieving Synchronization Logs* on page 125
- *Retrieving Device Notification Logs* on page 126
- *Retrieving Data Change Notification Logs* on page 127
- *Retrieving Security Logs* on page 128
- *Retrieving Error Logs* on page 129
- *Retrieving Connection Logs* on page 130
- *Retrieving Push Logs* on page 131
- *Retrieving Proxy Logs* on page 132
- *Retrieving Server Logs* on page 133
- *Retrieving Dispatcher Logs* on page 134
- *Correlating Log Data Across Subsystems* on page 136
- *Exporting Log Data* on page 137

Correlating Log Data Across Subsystems

Correlation mode enables you to retrieve domain log data in multiple subsystems, using the same search condition. The same condition is combined by common type filters and time range. This provides a tool for correlating activity across subsystems, useful for analyzing and troubleshooting.

For example, you could create a common filter for Application ID (such as Application ID = appid); select the Proxy tab and enter the filter information; retrieve the data; then select Correlated mode, and switch to another tab such as Proxy.

1. In the left navigation pane of SAP Control Center, select **Domain**.
2. Under the domain node, select **Log**.
3. In the right administration pane, select the **General** tab.
4. Select the subsystem tab, such as **Synchronization**.
5. To select or set up a special filter, select **Show filter**.
6. To filter based on the date and time, enter a **Start date**, **End date**, **Start time**, **End time**.
7. (Optional) Select the columns to display from the drop down list, such as Application ID, Application Connection Id, User, and Thread.
8. Click **Retrieve** to retrieve the logs.

The table is populated with log data.

9. Select the common type of log filters to use.
10. Switch to another subsystem tabs. The data is refreshed using the same criteria from the common type of log filters and time range specified.
11. Use the data to trace application activity across subsystems.

See also

- *Supported Log Subsystems* on page 120
- *Setting Up a Pool of Log Filters* on page 121
- *Retrieving Unified View Logs* on page 124
- *Retrieving Synchronization Logs* on page 125
- *Retrieving Device Notification Logs* on page 126
- *Retrieving Data Change Notification Logs* on page 127
- *Retrieving Security Logs* on page 128
- *Retrieving Error Logs* on page 129
- *Retrieving Connection Logs* on page 130
- *Retrieving Push Logs* on page 131
- *Retrieving Proxy Logs* on page 132
- *Retrieving Server Logs* on page 133
- *Retrieving Dispatcher Logs* on page 134
- *Retrieving Application Logs* on page 135
- *Exporting Log Data* on page 137

Exporting Log Data

Export data to a file for archive, or to analyze and troubleshoot problems.

Note:

- Depending on the amount of data being exported, the log export can take a long time. SAP recommends using log filters and time ranges to filter out and export specific log entries of interest.
 - You can also use the management API if there is a need to export domain log contents if the data set is large, or to refrain from blocking the user interface.
-

1. In the left navigation pane of SAP Control Center, select **Domains**.
2. Under the domain node, select **Log**.
3. In the right administration pane, select the **General** tab.
4. Select the subsystem tab, such as **Synchronization**.
5. To select or set up a special filter, select **Show filter**.

6. To filter based on the date and time, enter a **Start date**, **End date**, **Start time**, **End time**.
7. Click **Retrieve** to retrieve the logs.
The table is populated with log data.
8. Click **Export** and specify the following information:
 - a) To compress and reduce the size of the domain log, select **Compress the domain log**.
 - b) To set the number of log entries to retrieve per batch, set the **Log entry retrieve batch size**. Setting a large batch size big reduces the export time, but consumes more memory. The default batch size is 2000.
For example, if the total log entries based on the filter are 100,000, and the batch size is 10,000, then the export fetches 10 batches of log entries.
 - c) Set the domain log file name and location.

See also

- *Supported Log Subsystems* on page 120
- *Setting Up a Pool of Log Filters* on page 121
- *Retrieving Unified View Logs* on page 124
- *Retrieving Synchronization Logs* on page 125
- *Retrieving Device Notification Logs* on page 126
- *Retrieving Data Change Notification Logs* on page 127
- *Retrieving Security Logs* on page 128
- *Retrieving Error Logs* on page 129
- *Retrieving Connection Logs* on page 130
- *Retrieving Push Logs* on page 131
- *Retrieving Proxy Logs* on page 132
- *Retrieving Server Logs* on page 133
- *Retrieving Dispatcher Logs* on page 134
- *Retrieving Application Logs* on page 135
- *Correlating Log Data Across Subsystems* on page 136

Purging Domain Logs

A manual purge request is submitted to the primary server. It is a background task is initiated to perform batched clean-up of the data from the domain log database.

Note: Do not stop the primary server while the purge task is executing. Otherwise, you will need to submit the manual purge request again.

1. In the left navigation pane of SAP Control Center, select **Domains**.
2. Under the domain node, select **Log**.

3. In the right administration pane, select the **Settings** tab.
4. Click **Purge**.
5. Enter the date and time within which you want the data to be purged.
6. Click **OK**.

The purge completion time is dependent on the amount of the log data being purged. Therefore, domain log data may still be seen during this time.

Domain Log Categories

Domain log data provides detailed statistics for all aspects of device, user, application, domain, and data synchronization related activities.

Synchronization Log

Synchronization logs include data related to different aspects of data synchronization, including data, subscriptions, operations, result checker, cache refresh and the data service and SAP Mobile Server interface. Using data in these logs and the correlation tool, you can follow the data path between the enterprise information system (EIS), SAP Mobile Server, cache database, and user application connection.

To find out about	See
Data synchronization transactions	Data Sync statistics
Data services requests made to the Enterprise information system (EIS)	DS Interface statistics
Cache database (CDB) activities	Cache refresh statistics
EIS error codes or failures resulting from Mobile Business Object operations against the EIS data-source	Result Checker statistics (coding required)
Moving MBO operations from a mobile device to the CDB	Operation replay statistics
Moving data between a mobile device and the CDB	Subscription statistics

Data Sync

Synchronization logs include data related to different aspects of data synchronization, including data and SAP Mobile Server interface.

Data Sync – basic statistics for individual data synchronizations:

- Time – the time and date stamp for the log entry.
- Application ID – the unique identifier assigned to the registered application. Values may include a number, blank, or HWC, depending on the client type.

- Application Connection ID – the unique identifier for a user application connection.
- User ID – the ID of the user associated with the application ID. The user under the domain logs should be in the format "username@securityConfiguration".
- Stage – the current stage of processing - START or FINISH.
- Package – the name of the package to which the subscription belongs.
- MBO – the mobile business object used.
- Sync Group – the synchronization group associated with the request.
- Thread ID – the identifier for the thread used to process the request.
- Node ID – the server node on which the request is received.
- Error – the error message if any.
- Transaction ID – a unique ID that represents a transaction (one cycle of request-response) performed by the client or application.
- Root Context ID – a unique ID that represents a client/server session. A session can be thought of as a block that includes multiple requests from the client to the server.

Note: Additional detail columns:

- Payload
-

Operation Replay

Synchronization logs include data related to different aspects of data synchronization, including operations and SAP Mobile Server interface.

Operation Replay – statistics for moving MBO operations (typically create, update, and delete) from the device cache to the cache database cache on SAP Mobile Server:

- Time – the time and date stamp for the log entry.
- Application ID – the unique identifier assigned to the registered application. Values may include a number, blank, or HWC, depending on the client type.
- Application Connection ID – the unique identifier for a user application connection.
- User ID – the ID of the user associated with the application ID. The user under the domain logs should be in the format "username@securityConfiguration".
- Stage – the current stage of processing - START or FINISH.
- Package – the name of the package to which the subscription belongs.
- MBO – the mobile business object used.
- Operation – the MBO operation.
- Thread ID – the identifier for the thread used to process the request.
- Node ID – the server node on which the request is received.
- Error – the error message if any.
- Transaction ID – a unique ID that represents a transaction (one cycle of request-response) performed by the client or application.

- Root Context ID – a unique ID that represents a client/server session. A session can be thought of as a block that includes multiple requests from the client to the server.

Note: Additional detail columns:

- Payload
-

Subscription

Synchronization logs include data related to different aspects of data synchronization, including subscriptions and SAP Mobile Server interface.

Subscription – statistics for transferring data between mobile devices and the cache database on SAP Mobile Server:

- Time – the time and date stamp for the log entry.
- Application ID – the unique identifier assigned to the registered application. Values may include a number, blank, or HWC, depending on the client type.
- Application Connection ID – the unique identifier for a user application connection.
- User ID – the ID of the user associated with the application ID. The user under the domain logs should be in the format "username@securityConfiguration".
- Stage – the current stage of processing - START or FINISH.
- Package – the name of the package to which the subscription belongs.
- Subscription Type – the type of subscription used, including SUBSCRIBE, UNSUBSCRIBE, RECOVER, SUSPEND, and RESUME.
- Subscription ID – the identifier associated with the subscription.
- Sync Group – the synchronization group associated with the request.
- Thread ID – the identifier for the thread used to process the request.
- Node ID – the server node on which the request is received.
- Error – the error message if any.
- Transaction ID – a unique ID that represents a transaction (one cycle of request-response) performed by the client or application.
- Root Context ID – a unique ID that represents a client/server session. A session can be thought of as a block that includes multiple requests from the client to the server.

Note: Additional detail columns:

- Payload
-

Result Checker

Synchronization logs include data related to different aspects of data synchronization, including result checker and SAP Mobile Server interface.

Result Checker – EIS error codes or failures resulting from Mobile Business Object operations against the EIS datasource (requires coding):

- Time – the time and date stamp for the log entry.
- Application ID – the unique identifier assigned to the registered application. Values may include a number, blank, or HWC, depending on the client type.
- Application Connection ID – the unique identifier for a user application connection.
- User ID – the ID of the user associated with the application ID. The user under the domain logs should be in the format "username@securityConfiguration".
- Stage – the current stage of processing - START or FINISH.
- Package – the name of the package to which the subscription belongs.
- Class Name – the class used for the result checker.
- Thread ID – the identifier for the thread used to process the request.
- Node ID – the server node on which the request is received.
- Error – the error message if any.
- Transaction ID – a unique ID that represents a transaction (one cycle of request-response) performed by the client or application.
- Root Context ID – a unique ID that represents a client/server session. A session can be thought of as a block that includes multiple requests from the client to the server.

Note: Additional detail columns:

- None
-

Cache Refresh

Synchronization logs include data related to different aspects of data synchronization, including cache refresh and SAP Mobile Server interface.

Cache Refresh – statistics for cache database activities:

- Time – the time and date stamp for the log entry.
- Application ID – the unique identifier assigned to the registered application. Values may include a number, blank, or HWC, depending on the client type.
- Application Connection ID – the unique identifier for a user application connection.
- User ID – the ID of the user associated with the application ID. The user under the domain logs should be in the format "username@securityConfiguration".
- Stage – the current stage of processing - START or FINISH.
- Package – the name of the package to which the subscription belongs.
- MBO – the mobile business object used.
- Cache Group – the cache group name.
- CacheRow Count – the number of cached rows.
- EIS Row Count – the number of rows retrieved from the enterprise information system (EIS).
- Insert Count – the number of rows inserted in the cache.
- Update Count – the number of rows updated in the cache.

- Delete Count – the number of rows deleted from the cache.
- Thread ID – the identifier for the thread used to process the request.
- Node ID – the server node on which the request is received.
- Error – the error message if any.
- Transaction ID – a unique ID that represents a transaction (one cycle of request-response) performed by the client or application.
- Root Context ID – a unique ID that represents a client/server session. A session can be thought of as a block that includes multiple requests from the client to the server.

Note: Additional detail columns:

- Refresh Type
 - Virtual Table Name
 - Partition Key
 - Pre Update Cache Image (payload)
 - Post Update Cache Image (payload)
-

DS Interface

Synchronization logs include data related to different aspects of data synchronization, including data service and SAP Mobile Server interface.

DS Interface – statistics for data services requests made to the Enterprise information system (EIS):

- Time – the time and date stamp for the log entry.
- Application ID – the unique identifier assigned to the registered application. Values may include a number, blank, or HWC, depending on the client type.
- Application Connection ID – the unique identifier for a user application connection.
- User ID – the ID of the user associated with the application ID. The user under the domain logs should be in the format "username@securityConfiguration".
- Stage – the current stage of processing - START or FINISH.
- Package – the name of the package to which the subscription belongs.
- MBO – the mobile business object used.
- Operation – the MBO operation.
- Thread ID – the identifier for the thread used to process the request.
- Node ID – the server node on which the request is received.
- Error – the error message if any.
- Transaction ID – a unique ID that represents a transaction (one cycle of request-response) performed by the client or application.
- Root Context ID – a unique ID that represents a client/server session. A session can be thought of as a block that includes multiple requests from the client to the server.

Note: Additional detail columns:

- Operation Type
 - Virtual Table Name
 - Input Attributes (payload)
 - Input Parameters (payload)
-

Device Notification Log

Device notification logs include logging data for server-initiated synchronization notifications between SAP Mobile Server and devices.

- Time – the time and date stamp for the log entry.
 - Application ID – the unique identifier assigned to the registered application. Values may include a number, blank, or HWC, depending on the client type.
 - Application Connection ID – the unique identifier for a user application connection.
 - User ID – the ID of the user associated with the application ID. The user under the domain logs should be in the format "username@securityConfiguration".
 - Package – the name of the package to which the subscription belongs.
 - MBO – the mobile business object used.
 - Sync Group – the synchronization group associated with the request.
 - Thread ID – the identifier for the thread used to process the request.
 - Node ID – the server node on which the request is received.
 - Error – the error message if any.
 - Transaction ID – a unique ID that represents a transaction (one cycle of request-response) performed by the client or application.
 - Root Context ID – a unique ID that represents a client/server session. A session can be thought of as a block that includes multiple requests from the client to the server.
-

Note: Additional detail columns:

- Payload
-

Data Change Notification Log

Data Change Notification (DCN) logs include logging data for data change notifications between an enterprise information system (EIS) and an MBO package, for general and Hybrid App DCN.

General Data Change Notification

Provides logging data for general data change notifications between an enterprise information system (EIS) and an MBO package.

- Time – the time and date stamp for the log entry.
- User ID – the ID of the user associated with the application ID. The user under the domain logs should be in the format "username@securityConfiguration".

- Stage – the current stage of processing - START or FINISH.
- Package – the name of the package to which the subscription belongs.
- MBO – the mobile business object used.
- Thread ID – the identifier for the thread used to process the request.
- Node ID – the server node on which the request is received.
- Error – the error message if any.
- Transaction ID – a unique ID that represents a transaction (one cycle of request-response) performed by the client or application.
- Root Context ID – a unique ID that represents a client/server session. A session can be thought of as a block that includes multiple requests from the client to the server.

Note: Additional detail columns:

- Payload
-

Hybrid App Data Change Notification

Provides logging data for Hybrid App data change notifications between an enterprise information system (EIS) and an MBO package.

- Time – the time and date stamp for the log entry.
- Hybrid App ID – the unique identifier associated with a Hybrid App.
- User ID – the ID of the user associated with the application ID. The user under the domain logs should be in the format "username@securityConfiguration".
- Package – the name of the package to which the subscription belongs.
- Thread ID – the identifier for the thread used to process the request.
- Node ID – the server node on which the request is received.
- Operation – the MBO operation.
- Subject – the Hybrid App DCN request subject line.
- From – the "From" value for the Hybrid App DCN request.
- To – the "To" value for the Hybrid App DCN request.
- Body – the message body for the Hybrid App DCN request.
- Error – the error message if any.
- Transaction ID – a unique ID that represents a transaction (one cycle of request-response) performed by the client or application.
- Root Context ID – a unique ID that represents a client/server session. A session can be thought of as a block that includes multiple requests from the client to the server.

Note: Additional detail columns:

- Payload
-

Security Log

Security logs provide security details for individual applications, application connections, and users. Logs capture authentication failures and errors, and provide supporting information

that identifies request-response messaging, package and MBO details, security configuration, and the thread and node that attempted to process an authentication request.

- Time – the time and date stamp for the log entry.
- Application ID – the unique identifier assigned to the registered application. Values may include a number, blank, or HWC, depending on the client type.
- Application Connection ID – the unique identifier for a user application connection.
- User ID – the ID of the user associated with the application ID. The user under the domain logs should be in the format "username@securityConfiguration".
- Correlation ID – the unique ID associated with every request-response message pair.
- Package – the name of the package to which the subscription belongs.
- MBO – the mobile business object used.
- Security Configuration – the associated security configuration.
- Operation – the MBO operation used.
- Thread ID – the identifier for the thread used to process the request.
- Node ID – the server node on which the request is received.
- Outcome – the authentication outcome for the security check.
- Reason – the reason for authentication failure.
- Error – the error message if any.
- Transaction ID – a unique ID that represents a transaction (one cycle of request-response) performed by the client or application.
- Root Context ID – a unique ID that represents a client/server session. A session can be thought of as a block that includes multiple requests from the client to the server.

Error Log

Errors log data includes domain-level errors.

- Time – the time and date stamp for the log entry.
- Application ID – the unique identifier assigned to the registered application. Values may include a number, blank, or HWC, depending on the client type.
- Application Connection ID – the unique identifier for a user application connection.
- User ID – the ID of the user associated with the application ID. The user under the domain logs should be in the format "username@securityConfiguration".
- Correlation ID – the unique ID associated with every request-response message pair.
- Package – the name of the package to which the subscription belongs.
- MBO – the mobile business object used.
- Operation – the MBO operation.
- Thread ID – the identifier for the thread used to process the request.
- Node ID – the server node on which the request is received.
- Error – the error message if any.
- Transaction ID – a unique ID that represents a transaction (one cycle of request-response) performed by the client or application.

- Root Context ID – a unique ID that represents a client/server session. A session can be thought of as a block that includes multiple requests from the client to the server.

Connection Log

Connections log data includes domain connections for specific connection types to backend data sources, including DOE, JDBC, REST, SAP, and SOAP, if enabled. Check the detail pane for additional columns that may be available. Enable Payload to see payload data that may be available.

DOE Connection

Connections log data includes domain connections for DOE connection types, if enabled. Check the detail pane for additional columns that may be available. Enable Payload to see payload data that may be available.

- Time – the time and date stamp for the log entry.
- Application ID – the unique identifier assigned to the registered application. Values may include a number, blank, or HWC, depending on the client type.
- Application Connection ID – the unique identifier for a user application connection.
- User – the subscription user for the package.
- Event Type – the DOE-C event type, such as Acknowledged, Duplicate Ignored, Exclude, No Response (from client or server), Packet Dropped, Registration Response, Resend (from client), Status Request (from client or server), DOE-C Subscription, and DOE-C Data Import.
- Package – the name of the package to which the subscription belongs.
- MBO – the mobile business object used.
- Operation – the MBO operation.
- Connection – the managed connection used. Its value is DOE for DOE-C logs.
- Client ID – the identifier for the DOE-C client.
- Physical ID – the DOE-C generated physical identifier registered with DOE at subscription.
- Subscription ID – the DOE-C generated subscription identifier registered with DOE at subscription.
- Logical Device ID – the DOE-C logical device identifier, generated by DOE and provided to DOE-C upon successful subscription.
- Message Direction – the DOE-C message direction, either client to SAP Mobile Server, or SAP Mobile Server to client.
- Thread ID – the identifier for the thread used to process the request.
- Node ID – the server node on which the request is received.
- Error – the error message if any.

Note: Payload and detail columns:

- Device ID – the core and administrative (MMS) device ID.

- Domain – the core and administrative (MMS) domain name.
 - JSON Message Content – the messaging synchronization JSON message (payload). This is the SAP Mobile Platform-specific representation of the incoming DOE XML message in JSON format. DOE-C receives XML the payload from DOE in response, which is then parsed and converted to a JSON string and sent to the client.
 - XML Message Content – the DOE SOAP messages (payload). This represents either an XML request in a format for sending to DOE by DOE-C, or an XML payload response received from DOE as applicable.
 - Endpoint Name – the core and administrative (MMS) endpoint name.
 - DOE server message ID – the SAP DOE reliable messaging server message ID.
 - DOE client message ID – the SAP DOE reliable messaging client message ID.
 - DOE-C server message ID – the DOE-C client-side SAP DOE reliable messaging server message ID.
 - DOE-C client message ID – the DOE-C client-side SAP DOE reliable messaging client message ID.
 - DOE-C method name – the DOE-C method being executed.
 - DOE-C action name – the DOE SOAP action.
 - Push to – the messaging asynchronous response queue.
 - Address – the remote URL of the DOE server for this subscription (for example, `http://saphost:50015/sap/bc/DOE_ESDMA_SOAP?sap-client=600`).
 - Log – the DOE-C subscription-specific log level.
 - Extract Window – the DOE extract window for a subscription. This value determines the maximum number of unacknowledged "in-flight" messages allowed by the DOE reliable messaging protocol.
 - PBI – the messaging synchronization "piggy backed import" setting for the subscription.
 - Boolean property – indicates whether replay after-images can be piggy-backed onto `replayResult` and `replayFailed` messages (default is false).
-

JDBC Connection

Connections log data includes domain connections for JDBC connection types to backend data sources, if enabled. Check the detail pane for additional columns that may be available. Enable Payload to see payload data that may be available.

- Time – the time and date stamp for the log entry.
- Application ID – the unique identifier assigned to the registered application. Values may include a number, blank, or HWC, depending on the client type.
- Application Connection ID – the unique identifier for a user application connection.
- User ID – the ID of the user associated with the application ID. The user under the domain logs should be in the format "username@securityConfiguration".
- Stage – the current stage of processing - START or FINISH.
- Package – the name of the package to which the subscription belongs.

- MBO – the mobile business object used.
- Operation – the MBO operation.
- Connection – the managed connection used.
- Thread ID – the identifier for the thread used to process the request.
- Node ID – the server node on which the request is received.
- Error – the error message if any.
- Transaction ID – a unique ID that represents a transaction (one cycle of request-response) performed by the client or application.
- Root Context ID – a unique ID that represents a client/server session. A session can be thought of as a block that includes multiple requests from the client to the server.

Note: Payload and detail columns:

- Input Parameters – the parameters used in a JDBC endpoint operation (payload). This will vary by operation.
 - Query – the SQL statement used in a JDBC endpoint operation (payload). This will vary by operation.
 - Device ID – the core and administrative (MMS) device ID.
 - Domain – the core and administrative (MMS) domain name.
 - Endpoint Name – the core and administrative (MMS) endpoint name.
 - Database Product Name – the remote database product name, such as "SQL Anywhere".
 - Database Product Version – the remote database version, such as "11.0.1.2044".
 - Driver Name – the database driver used, such as: "jConnect™ for JDBC™".
 - Driver Version – the database driver version, such as "jConnect™ for JDBC™/7.07 GA(Build 26666)/P/EBF19485/JDK 1.6.0/jdbcmain/Wed Aug 31 03:14:04 PDT 2011".
 - Database User Name – the database user account.
-

REST Connection

Connections log data includes domain connections for REST connection types to backend data sources, if enabled. Check the detail pane for additional columns that may be available. Enable Payload to see payload data that may be available.

- Time – the time and date stamp for the log entry.
- Application ID – the unique identifier assigned to the registered application. Values may include a number, blank, or HWC, depending on the client type.
- Application Connection ID – the unique identifier for a user application connection.
- User ID – the ID of the user associated with the application ID. The user under the domain logs should be in the format "username@securityConfiguration".
- Stage – the current stage of processing - START or FINISH.
- Package – the name of the package to which the subscription belongs.
- MBO – the mobile business object used.
- Operation – the MBO operation.

- Connection – the managed connection used.
- URL – the URL associated with the managed connection.
- Action – the GET, POST, PUT, or DELETE action.
- Response Status – the response status code for the invocation.
- Thread ID – the identifier for the thread used to process the request.
- Node ID – the server node on which the request is received.
- Error – the error message if any.
- Transaction ID – a unique ID that represents a transaction (one cycle of request-response) performed by the client or application.
- Root Context ID – a unique ID that represents a client/server session. A session can be thought of as a block that includes multiple requests from the client to the server.

Note: Payload and detail columns:

- Response – the message returned by the EIS system in response to a request (payload).
 - Device ID – the core and administrative (MMS) device ID.
 - Domain – the core and administrative (MMS) domain name.
 - Endpoint Name – the core and administrative (MMS) endpoint name.
 - HTTP Header Parameters – "Accept-Encoding: gzip, Accept-Encoding: compress".
-

SAP Connection

Connections log data includes domain connections for SAP connection types to backend data sources, if enabled. Check the detail pane for additional columns that may be available. Enable Payload to see payload data that may be available.

- Time – the time and date stamp for the log entry.
- Application ID – the unique identifier assigned to the registered application. Values may include a number, blank, or HWC, depending on the client type.
- Application Connection ID – the unique identifier for a user application connection.
- User ID – the ID of the user associated with the application ID. The user under the domain logs should be in the format "username@securityConfiguration".
- Stage – the current stage of processing - START or FINISH.
- Package – the name of the package to which the subscription belongs.
- MBO – the mobile business object used.
- Operation – the MBO operation.
- BAPI – the SAP BAPI used as the data source.
- Connection – the managed connection used.
- Properties – the list of name:value pairs.
- Thread ID – the identifier for the thread used to process the request.
- Node ID – the server node on which the request is received.
- Error – the error message if any.

- Transaction ID – a unique ID that represents a transaction (one cycle of request-response) performed by the client or application.
- Root Context ID – a unique ID that represents a client/server session. A session can be thought of as a block that includes multiple requests from the client to the server.

Note: Payload and detail columns:

- Parameters – input that was supplied to the operation; this will vary per request and operation (payload).
 - Device ID – the core and administrative (MMS) device ID.
 - Domain – the core and administrative (MMS) domain name.
 - Endpoint Name – the core and administrative (MMS) endpoint name.
 - SAP Host – the remote system hostname (if available).
 - SAP User – the SAP user for the operation.
-

SOAP Connection

Connections log data includes domain connections for SOAP connection types to backend data sources, if enabled. Check the detail pane for additional columns that may be available. Enable Payload to see payload data that may be available.

- Time – the time and date stamp for the log entry.
- Application ID – the unique identifier assigned to the registered application. Values may include a number, blank, or HWC, depending on the client type.
- Application Connection ID – the unique identifier for a user application connection.
- User ID – the ID of the user associated with the application ID. The user under the domain logs should be in the format "username@securityConfiguration".
- Stage – the current stage of processing - START or FINISH.
- Package – the name of the package to which the subscription belongs.
- MBO – the mobile business object used.
- Operation – the MBO operation.
- Connection – the managed connection used.
- Service Address – the service address URL.
- Action – the SOAP action.
- Thread ID – the identifier for the thread used to process the request.
- Node ID – the server node on which the request is received.
- Error – the error message if any.
- Transaction ID – a unique ID that represents a transaction (one cycle of request-response) performed by the client or application.
- Root Context ID – a unique ID that represents a client/server session. A session can be thought of as a block that includes multiple requests from the client to the server.

Note: Payload and detail columns:

- Request (payload) – SOAP messages sent to the remote SOAP service.
 - Response (payload) – SOAP messages received from the remote SOAP service.
 - Device ID – the core and administrative (MMS) device ID.
 - Domain – the core and administrative (MMS) domain name.
 - Endpoint Name – the core and administrative (MMS) endpoint name.
 - Connection Timeout – the response timeout window, in milliseconds.
 - Authentication Type – the authentication type, either "None", "Basic", "SSO2", or "X509".
-

Push Log

Push logs include log data for all push notifications.

- Time – the time and date stamp for the log entry.
- Application ID – the unique identifier assigned to the registered application. Values may include a number, blank, or HWC, depending on the client type.
- Application Connection ID – the unique identifier for a user application connection.
- User ID – the ID of the user associated with the application ID. The user under the domain logs should be in the format "username@securityConfiguration".
- Source - the source of the log if its from the server or client.
- Correlation ID - the unique id associated with every request-response message pair.
- URN - not relevant.
- Log Level - not relevant.
- Thread ID – the identifier for the thread used to process the request.
- Node ID – the server node on which the request is received.
- Error – the error message if any.
- Transaction ID – a unique ID that represents a transaction (one cycle of request-response) performed by the client or application.
- Root Context ID – a unique ID that represents a client/server session. A session can be thought of as a block that includes multiple requests from the client to the server.
- Device Type – device type from which the push message originated.
- Notification Type – type of notification. For example Native.
- Received Time – a time stamp indicating when the message was received by the server.
- Processing Started Time – a time stamp indicating when the server started processing the message.

Proxy Log

Proxy logs all data made to and from the Proxy server.

- Time – the time and date stamp for the log entry.
- Application ID – the unique identifier assigned to the registered application. Values may include a number, blank, or HWC, depending on the client type.

- Application Connection ID – the unique identifier for a user application connection.
- User ID – the ID of the user associated with the application ID. The user under the domain logs should be in the format "username@securityConfiguration".
- Phase – as an aid in evaluating network performance, as part of overall application performance, select a phase to measure time of these events:
 - **FBI** – (first byte in) the time that the first byte is read from the request.
 - **LBI** – (last byte in) the time that the last byte is read from the request.
 - **FBO** – (first byte out) the time that the first byte is written to the response.
 - **LBO** – (last byte out) the time that the last byte is written to the response.
- Source - the source of the log if its from the server or client.
- Correlation ID - the unique id associated with every request-response message pair.
- Request Type - the request type of the message.
- Request URL - the Gateway URL.
- HTTP Endpoint - the Gateway URL.
- Response Code – the response status code for the invocation.
- Log Level - not relevant.
- Thread ID – the identifier for the thread used to process the request.
- Node ID – the server node on which the request is received.
- Transaction ID – a unique ID that represents a transaction (one cycle of request-response) performed by the client or application.
- Root Context ID – a unique ID that represents a client/server session. A session can be thought of as a block that includes multiple requests from the client to the server.

Note: Additional detail columns:

- Post Data
 - Request Header Fields
 - Response Body
 - Response Header Fields
-

Server Log

Server logs include logging data for SAP Mobile Server.

- Time – the time and date stamp for the log entry.
- Application ID – the unique identifier assigned to the registered application. Values may include a number, blank, or HWC, depending on the client type.
- Application Connection ID – the unique identifier for a user application connection.
- User ID – the ID of the user associated with the application ID. The user under the domain logs should be in the format "username@securityConfiguration".
- Package – the name of the package to which the subscription belongs.
- Correlation ID – the unique id associated with the correlated data. see *Correlating Log Data Across Subsystems* for details. For example, root context id and Transaction id

appear in every trace entry, and contain the values used to correlate trace entries from different subsystems (buckets).

- Log Level – indicates the log level, if any, set on the client that controls what and how much should be logged.
- Thread ID – the identifier for the thread used to process the request.
- Node ID – the server node on which the request is received.
- Bucket – the subsystem from which the logging data originates. For example Security or Data Services (DS).
- Category – the category or type of information under which the data is logged.
- Transaction ID – a unique ID that represents a transaction (one cycle of request-response) performed by the client or application.
- Root Context ID – a unique ID that represents a client/server session. A session can be thought of as a block that includes multiple requests from the client to the server.

Dispatcher Log

Dispatcher log data includes dispatcher specific messages for Messaging and Service.

Messaging Log

Messaging log data includes data for all message-based application data routed by the dispatcher.

- Time – the time and date stamp for the log entry.
- Application ID – the unique identifier assigned to the registered application. Values may include a number, blank, or HWC, depending on the client type.
- Application Connection ID – the unique identifier for a user application connection.
- User ID – the ID of the user associated with the application ID. The user under the domain logs should be in the format "username@securityConfiguration".
- Phase – as an aid in evaluating network performance, as part of overall application performance, select a phase to measure time of these events:
 - **FBI** – (first byte in) the time that the first byte is read from the request.
 - **LBI** – (last byte in) the time that the last byte is read from the request.
 - **FBO** – (first byte out) the time that the first byte is written to the response.
 - **LBO** – (last byte out) the time that the last byte is written to the response.
- Source - the source of the log if its from the server or client.
- Request URL - the Gateway URL.
- Response Code – the response status code for the invocation.
- Log Level – indicates the log level, if any, set on the client that controls what and how much should be logged.
- Thread ID – the identifier for the thread used to process the request.
- Node ID – the server node on which the request is received.

- Transaction ID – a unique ID that represents a transaction (one cycle of request-response) performed by the client or application.
- Root Context ID – a unique ID that represents a client/server session. A session can be thought of as a block that includes multiple requests from the client to the server.
- Request Header Fields - the HTTP request header field contained in the application. For example, used by REST API-based application to create an application connection.
- Response Header Fields - the HTTP response header field communicated to the device from the server.

Service Log

Service log data includes application service data routed by the dispatcher.

- Time – the time and date stamp for the log entry.
- Application ID – the unique identifier assigned to the registered application. Values may include a number, blank, or HWC, depending on the client type.
- Application Connection ID – the unique identifier for a user application connection.
- User ID – the ID of the user associated with the application ID. The user under the domain logs should be in the format "username@securityConfiguration".
- Phase – as an aid in evaluating network performance, as part of overall application performance, select a phase to measure time of these events:
 - **FBI** – (first byte in) the time that the first byte is read from the request.
 - **LBI** – (last byte in) the time that the last byte is read from the request.
 - **FBO** – (first byte out) the time that the first byte is written to the response.
 - **LBO** – (last byte out) the time that the last byte is written to the response.
- Source - the source of the log if its from the server or client.
- Response Code – the response status code for the invocation.
- Log Level – indicates the log level, if any, set on the client that controls what and how much should be logged.
- Thread ID – the identifier for the thread used to process the request.
- Node ID – the server node on which the request is received.
- Transaction ID – a unique ID that represents a transaction (one cycle of request-response) performed by the client or application.
- Root Context ID – a unique ID that represents a client/server session. A session can be thought of as a block that includes multiple requests from the client to the server.
- Request Header Fields - the HTTP request header field contained in the application. For example, used by REST API-based application to create an application connection.
- Response Header Fields - the HTTP response header field communicated to the device from the server.

Application Log

Application log data includes application specific messages, including Registration and Setting.

Registration Log

Registration log data includes registration-related application data. The registration log applies only to applications registered through the HTTP channel.

- Time – the time and date stamp for the log entry.
- Application ID – the unique identifier assigned to the registered application. Values may include a number, blank, or HWC, depending on the client type.
- Application Connection ID – the unique identifier for a user application connection.
- User ID – the ID of the user associated with the application ID. The user under the domain logs should be in the format "username@securityConfiguration".
- Source - the source of the log if its from the server or client.
- Log Level – indicates the log level, if any, set on the client that controls what and how much should be logged.
- Thread ID – the identifier for the thread used to process the request.
- Node ID – the server node on which the request is received.
- Transaction ID – a unique ID that represents a transaction (one cycle of request-response) performed by the client or application.
- Root Context ID – a unique ID that represents a client/server session. A session can be thought of as a block that includes multiple requests from the client to the server.
- Auto Registraton - the automatic connection registration setting of the application, as determined by the `autoreghint` provisioning property.
- Security Configuration - the security configuration assigned to the application.
- Template - the application template used by the application.

Setting Log

Setting log data includes application settings information.

- Time – the time and date stamp for the log entry.
- Application ID – the unique identifier assigned to the registered application. Values may include a number, blank, or HWC, depending on the client type.
- Application Connection ID – the unique identifier for a user application connection.
- User ID – the ID of the user associated with the application ID. The user under the domain logs should be in the format "username@securityConfiguration".
- Phase – as an aid in evaluating network performance, as part of overall application performance, select a phase to measure time of these events:
 - **FBI** – (first byte in) the time that the first byte is read from the request.
 - **LBI** – (last byte in) the time that the last byte is read from the request.
 - **FBO** – (first byte out) the time that the first byte is written to the response.
 - **LBO** – (last byte out) the time that the last byte is written to the response.
- Source - the source of the log if its from the server or client.

- Log Level – indicates the log level, if any, set on the client that controls what and how much should be logged.
- Thread ID – the identifier for the thread used to process the request.
- Node ID – the server node on which the request is received.
- Transaction ID – a unique ID that represents a transaction (one cycle of request-response) performed by the client or application.
- Root Context ID – a unique ID that represents a client/server session. A session can be thought of as a block that includes multiple requests from the client to the server.
- Operation – the MBO operation.
- Request Body - the application's HTTP request body field.
- Response Body - the server's HTTP response body field.

Agentry Application Logs

Agentry application logs record events and client-server messages for Agentry applications.

Events.log File

The `events.log` file records various Agentry Server events that occur at run time. Events include startup and shutdown, opening and closing connections to the application's system connections, loading of Client-Server communication components, memory usage, thread expansion, and used and available storage.

The events log persists until one of the following occurs:

- The log files are rolled by the Server, at which point they are moved to a backup folder named `Logs-Rolled`, in a sub-directory named for the date and time. By default, this occurs once every 24 hours provided the Agentry Server is running. With a change to the Server's configuration, this file may also be moved to a back up location based on file size.
- The Agentry Server is shut down and restarted. In this case, the events log file is moved to the backup folder `Logs-Rolled`, again in a sub-directory named for the date and time.

Unlike most other log files generated by the Agentry Server, the events log is always enabled. It cannot be disabled.

The following is an example of the contents of the `events.log` file:

```
11/28/2005 15:11:01, 1, 3, 7, Thr 1912, Server, Agentry Configuration,
  userKey, aaaa, File: agentry.ini,
  C:\Syclo\agentry\system.cpp#2103:DTLoggerHandler::badKey
11/28/2005 15:11:01, 0, 0, 2, Thr 1912, System Startup
11/28/2005 15:11:01, 1, 3, 8, Thr 1912, Invalid server license: Invalid User
  Key, C:\Syclo\agentry\server.cpp#444:System Startup
11/28/2005 15:11:01, 0, 11, 14, Thr 1912, TCP v4.0.0.0
11/28/2005 15:11:01, 0, 13, 14, Thr 1912, MID v4.0.0.0
11/28/2005 15:11:01, 0, 1, 4, Thr 1912, Agentry v4.0.0.0
```

Note: Each log message is written to a single line. Space restrictions in the example require some of the messages to wrap to multiple lines. Items in the example that are a part of the message above it are tabbed in. Each message begins with a timestamp.

The first column of each message contains the date and time of the event being logged. The next three columns, separated by commas, are the message type, message group, and the message ID. The message type is always a **0** or a **1**. A message type of **1** indicates an error message, while a message type of **0** indicates an informational message. The message type is followed by the thread ID, denoted by the text `Thr`, and finally the message itself.

Events Log Grid View

In the Grid View there are the following columns:

- Time: The timestamp of the message
- Message Type: The message type, either informational (0) or error (1).
- Thread ID L The processing thread from which the message was generated.
- Message: The log message content.

Messages.log File

The `messages.log` file contains log items related to client-server messaging. Each message and response sent between the Agentry Server and Agentry Client is a log item in the `messages.log` file. The information in each item includes the date and time of the message, the Agentry Client's IP address, the type of message or response, and other message information.

The `messages.log` persists until one of the following occurs:

- The log files are rolled by the Agentry Server, at which point they are moved to a backup folder named `Logs-Rolled`, in a sub-directory named for the date and time. By default, this occurs once every 24 hours provided the Agentry Server is running. With a change to the Agentry Server's configuration, this file may also be moved to a back up location based on file size.
- The Agentry Server is shut down and restarted. In this case, the `messages.log` file is moved to the backup folder `Logs-Rolled`, in a sub-directory named for the date and time.

Unlike most other log files generated by the Agentry Server, the `messages.log` file is always generated by the Agentry Server and cannot be disabled.

Messages Log Format

The following is an excerpt from a typical `messages.log` file in the Text View, with each column of output labeled. These same columns are also presented in the Grid View. See the list of columns below the example. where there is a description of each output column.

A,	2,	1,	99,	01/11/2012 09:04:55,	,	0,	6,	king,	ANGEL: A7CCE4F2-BBE4-421C-8CBB-BDFF35ECEDFE
S,	2,	1,	99,	01/11/2012 09:04:55,	0,	2,	6,	king,	ANGEL: A7CCE4F2-BBE4-421C-8CBB-BDFF35ECEDFE
C,	2,	1,	99,	01/11/2012 09:04:55,	,	2,	6,	king,	>unknown<
I,	9,	1,	100,	01/11/2012 09:04:58,	,	0,	125,	,	ANGEL: CA5CCA92-21FB-4B4E-9A29-9963F9250833
Q,	9,	1,	100,	01/11/2012 09:04:58,	,	0,	125,	king,	ANGEL: CA5CCA92-21FB-4B4E-9A29-9963F9250833
A,	9,	1,	100,	01/11/2012 09:04:58,	,	0,	125,	king,	ANGEL: CA5CCA92-21FB-4B4E-9A29-9963F9250833
S,	9,	1,	100,	01/11/2012 09:04:58,	16,	3,	125,	king,	ANGEL: CA5CCA92-21FB-4B4E-9A29-9963F9250833
I,	202,	20,	101,	01/11/2012 09:04:58,	,	0,	1815,	,	ANGEL: CA5CCA92-21FB-4B4E-9A29-9963F9250833
Q,	202,	20,	101,	01/11/2012 09:04:58,	,	0,	1815,	king,	ANGEL: CA5CCA92-21FB-4B4E-9A29-9963F9250833
C,	9,	1,	100,	01/11/2012 09:04:58,	,	3,	125,	king,	>unknown<
A,	202,	20,	101,	01/11/2012 09:04:58,	,	0,	1815,	king,	ANGEL: CA5CCA92-21FB-4B4E-9A29-9963F9250833
S,	202,	20,	101,	01/11/2012 09:04:58,	0,	2,	1815,	king,	ANGEL: CA5CCA92-21FB-4B4E-9A29-9963F9250833

- **Message State:** A single character indicating the state of the message in relation to the Agency Server. See the *Message States* and *Push Message States* tables later in this section for more information.
- **Message Code:** A numeric value that identifies the type of message. See the table *Message Codes* later in this section for more information.
- **Subsystem Code:** Always either a 1 or 20. This is a deprecated value that will only vary for older SAP products built on Agency 2.0 code libraries.
- **Message Number:** A number assigned by the Agency Client that increments for each message sent.
- **Date and Time:** The date and time the message was received by the Agency Server. This is always the date and time of the Agency Server's host system.
- **Response Code:** A numeric code that indicates the response to the message. The message codes "S" and "R" each have a different set of possible response codes. For all other message codes, the response code column is blank.
- **Data Sent/Data Received:** Specifies a running total of the amount of data sent and received by the Agency Server per message, with a value in bytes.
- **User ID:** Client ID of the user to which the message belongs. It is normal for the user ID to remain blank for messages with a message state of "I".
- **Client Location:** The IP address of the client device. It is normal for the Client Location to be unknown for messages with a message state of "C".

The following tables contain message state information for the `messages.log` file. There are two tables of message states. The first table is for synchronous transmissions. The second table is for asynchronous push messages only.

Table 9. Message States - Synchronous Communications

State	Name	Description
I	Incoming	Message in the process of being received from the client.
Q	Queued	Message is decoded, the user is identified, and the message is placed in one of the Agency Server's work queues.
A	Active	A previously queued message is now being processed by the Agency Server.
S	Sent Response	The Agency Server sent information and/or an acknowledgement to the Client.
R	Received Response	A Client response was received by the Agency Server.
C	Complete	The message processing is complete.

Table 10. Push Message States - Asynchronous Communications

State	Name	Description
O	Outgoing	A message is in the process of being sent to the Client.
T	Trying	The Agency Server is attempting to connect to a Client
L	Linked	The Agency Server has successfully connected to a Client.
W	Waiting	The Agency Server failed in an attempt to connect to a Client. It will attempt the connection again.
R	Received Response	The Agency Server has received a Client response.
S	Sent Response	The Agency Server sent information and/or an acknowledgement to the Client.
C	Complete	The message processing is complete.
X	Cancelled	The Agency Server cancelled the message to the Client.
F	Failed	The message has failed. The Agency Server will not retry.

The following table lists the message codes, which indicate the type of message being processed or sent. These values are found in the Message Code column of the `messages.log` file.

Table 11. Log Message Codes

Code	Description
2	Client logout request
3	Client login request
7	Client user password change request
200	Transaction instance sent to the Agentry Server from the Client
201	Client has made a fetch processing request
202	Client system information message sent to Agentry Server
203	Client has requested an object be reloaded/refreshed
204	Client has requested the definition of an object be sent
205	Client has requested the definition of a fetch be sent
206	Client has requested the definition of a transaction be sent
207	Client has requested the definition of a screen set be sent
208	Client has requested the definition of an action be sent
209	Client has requested the definition of a rule be sent
210	Client has requested the definition of a report be sent
211	Agentry Server has pushed an object and/or messages to the Client
212	Client has sent an enable push message to the Agentry Server for the user
213	Client has requested the definition of a style be sent
622	Client has requested all complex tables be updated
623	Client has requested all data tables be updated

Rolling Agentry Message and Event Log Files

Each Agentry application has its own Agentry Server instance that runs on the SAP Mobile Server node. Each Agentry Server instance generates message and event logs for the application, which can be viewed through the Domains folder in SAP Control Center. Agentry message and event log files are automatically rolled based on the configuration of the Agentry application server instance. You can also manually roll the log files.

When you roll an Agentry application log file, the SAP Mobile Server renames the `messages.log` and `events.log` based on the current date and time, and copies them to the backup folders `mlog` and `elog`, respectively. The log files are located in `SMP_HOME\Servers\UnwiredServer\logs\<agentryApplicationID>`.

1. In the navigation pane of SAP Control Center, expand the **Applications** node and select the application.
2. In the administration pane, click the **Logs** tab.
3. Click **Roll log files**.
4. Click **Yes** to confirm.

Client Logs

Client logs capture data about client application operations for all devices that are subscribed to a package. You can use the data to track errors and identify performance issues.

See also

- *Creating and Enabling a New Domain* on page 101
- *Deleting a Domain* on page 102
- *Registering a Domain Administrator User* on page 103
- *Assigning Domain Administrators to a Domain* on page 104
- *Viewing Applications for a Domain* on page 105
- *Viewing Application Connections for a Domain* on page 105
- *Scheduling Accumulated Data Cleanup for Domains* on page 107
- *Domain Logs* on page 110
- *Connections* on page 167
- *Configuring Domain Security* on page 195

Checking Client Application Logs

Review data about client application operations for all devices that are subscribed to a package in order to track errors and identify performance issues.

1. In the left navigation pane, click **Domains** > *<domainName>* > **Packages** and select the package you want to manage.
2. In the right administration pane, select the **Client Log** tab.
3. Review this information to monitor client application activity:
 - User – the name of the user that activates the device. The user under the client logs should be in the format "username@securityConfiguration".
 - Application Connection ID – the unique identifier for a user application connection.
 - MBO – the mobile business object that the client is synchronizing with.
 - Operation – the operation that the client is performing.
 - Code – the result of server-side operations; either 200 (successful) or other values (failed).
 - Level – the log level for the application; either FATAL, ERROR, WARN, INFO, DEBUG, or TRACE.
 - Timestamp – the date and time at which the operation took place.

- Message – the log message associated with the operation.
4. Select a row from the table and click **Details** to see a detailed view of data for the selected client log event.
 5. Click **Close** to return to the Client Log summary view.

Cleaning the Client Log

Keep client log records at a reasonable level, based on a retention period that is agreed upon by both the SAP Mobile Server administrator and the application developer. Clean the client log when there are log records that are older than the retention period.

The client log contains two types of logs:

- **Operation replay logs** – created on the SAP Mobile Server when an operation replay fails or results in a warning message. These log records are downloaded to the device when the application synchronizes.
- **Client logs** – created on the device by the application and uploaded to the server where they are retained for a period of time for audit or troubleshooting purposes.

Both types of logs are stored on the client device.

Cleaning the client log on the server involves these stages:

1. Mark the log records for deletion on the server (logical deletion).
2. Wait sufficient time for the client applications to synchronize with the SAP Mobile Server and remove the deleted records from the device database.

Note: The time allowed should take the user environment into account; for example, a user in a remote location may only be able to connect and synchronize once a week.

3. Delete the log records from the server (physical deletion).

When cleaning the client log, keep these best practices in mind:

- Perform the logical and physical cleanup of log records when client activity is at its lowest level to minimize the affect on client synchronization. Clients synchronizing when a purge is in process may encounter delays.
- Consider the requirements of all applications that use the MBOs in the mobile application package.

Marking Client Log Records for Deletion (Logical Deletion)

The first step in deleting client log records is to mark the log records for deletion (known as a logical deletion).

1. In the left navigation pane, expand the **Packages** folder and select the package you want to manage.
2. In the right administration pane, select the **Client Log** tab.
3. Click **Clean** or **Purge**.

4. Enter a time frame to indicate which client log data you want to mark for deletion, and click **OK**.

The data is marked for deletion, and no longer appears on the Client Log tab.

Next

When the client application synchronizes, the marked records are deleted from the client database. After synchronization, you can physically delete the records from the client log.

Deleting Client Log Records

Delete client records from the database by performing a cleanup. This physically delete records that have already been logically deleted. Logically deleted records are not physically deleted until the client application has synchronized the logical deletion. For example, if a record was logically deleted at 1:00 am on February 27, performing a cleanup deletes the log records for client applications that synchronized after that time.

You can clean up log records manually or schedule an automatic cleanup. You can also manually run the scheduled cleanup.

Note: Clean log records when client activity is at its lowest level. Clients cannot synchronize while a cleanup transaction is executing.

Scheduling Accumulated Data Cleanup for Domains

Periodically clean up accumulated data maintenance items in cache that are no longer needed, by creating a data purge schedule. Set the schedule to perform the clean up processes when system usage is low.

Alternatively, manually purge accumulated data items at any time; however, if the number of domain log records is very large (for example, hundreds of thousands) then SAP Mobile Server must purge domain log data asynchronously.

Note: You can use the Administration API to automate cleanup at the package level.

1. In the SAP Control Center left navigation pane, expand the **Domains** tab and select a domain.
2. In the right pane, select the **Scheduled Tasks** tab.
3. Under Task, select one of the options you want to schedule, and then select **Properties** to set up its automatic schedule:

Option	Description
Subscription Cleanup	Removes subscriptions that are inactive for the number of days specified in the schedule task configuration. A subscription is considered active based on: • Replication – last synchronization request time-stamp. • Messaging – last synchronization message time-stamp. A user who infrequently accesses the system, for example, 3-4 times a year, falls outside the specified time frame and is removed from SAP Mobile Platform. He or she must then reinstall the application and initiate a sync to reactivate the subscription.
Error History Cleanup	Removes historical data on MBO data refresh and operation replay failures, which result from system or application failures. System failures may include network problems, credential issues, and back-end system failure. Application failures may include invalid values, and non-unique data. Only error messages are removed.
Client Log Cleanup	Removes logically deleted client log records that have already been synchronized to the device, or are no longer associated with active users.
Synchronization Cache Cleanup	This cleanup task removes: • Logically deleted rows in the cache that are older than the oldest synchronization time on record in the system. Synchronization activity for all clients establish the oldest synchronization time. Note: The last synchronization time of a client is the last time the client successfully applied the download. Because the SAP Mobile Server does not require download acknowledgement, the SAP Mobile Server only knows that the previous download was applied successfully when the client synchronizes for the second time (uploading the previous synchronization time). The synchronization time on record in the system is normally the time the 'second last' synchronization happened. • Unused or stale partitions. In addition, the optional exp header property in DCN messages indicates an expiration date, after which the data is purged based on the defined Synchronization Cache Cleanup schedule.

4. Click **Properties**, then in the Task Properties dialog, select the **Schedule** tab.
5. Configure the required schedule:
 - **Schedule repeat** – select how often the schedule should run. Options are **daily**, **hourly**, **custom**, and **never**.
 - If you select **daily** or **hourly**, specify:
 - **Start date** – select the date and time the automated upload should begin. Use the calendar picker and 24-hour time selector.
 - **End date** – select the date and time the automated upload should end.
 - **Days of the week** – select each day the automated upload schedule should run.
 - Select **custom**, to specify the interval granularity in seconds, minutes, or hours, as well as other date and time parameters.
6. In the Task Properties dialog, select the **Options** tab, and set the number of inactive days for which to purge.
 For Subscription Cleanup, Error History Cleanup, and Client Log Cleanup, set the number of inactive days for which to purge. For Synchronization Cache Cleanup, set the purge interval for EIS managed caches. For more information on EIS-managed caches, see *EIS Defined Cache Partitions* in *Mobile Data Models: Using Mobile Business Objects* and *Creating Cache Groups* in *SAP Mobile Workspace - Mobile Business Object Development*.
7. Click **OK** to save the schedule properties and purge options.
8. Select **Enable**. Schedules run until you disable them, or until they expire.

Running Manual Cleanup by Domain

At any time the SUP Administrator or SUP Domain Administrator can manually run cleanup options. The processes run asynchronously on SAP Mobile Server using the current settings.

As much as is reasonable, use manual cleanup when system load is light.

1. In the left pane, expand the **Domains** folder, and select a domain.
2. In the right pane, select the **Scheduled Tasks** tab.
3. Select one of the cleanup options.
4. Click **Run Now**, then optionally specify the purge options.
 For Subscription Cleanup, Error History Cleanup, and Client Log Cleanup, set the number of inactive days for which to purge. For Synchronization Cache Cleanup, set the purge interval for EIS managed caches. For more information on EIS-managed caches, see *EIS Defined Cache Partitions* in *Mobile Data Models: Using Mobile Business Objects* and *Creating Cache Groups* in *SAP Mobile Workspace - Mobile Business Object Development*.
5. Click **OK (Yes)** to confirm. The request is sent immediately, and the task runs asynchronously on SAP Mobile Server.

Running Manual Cleanup By Package

Clean up client log records that have already been logically deleted and synchronized to the client application.

1. In the left navigation pane, open the **Domains** > *<domainName>* > **Packages** > *<packageName>*, then select the **General** tab in the administration pane.
2. Click **Error Cleanup**.

Connections

Connections allow SAP Mobile Server to communicate with data sources. To facilitate the connection process, define a set of properties for each data source. Establish connections and connection pools for each domain.

A connection is required to send queries to mobile business objects, and to receive answers. The format in which data is communicated depends on the type of data source; for example, database data sources use a result set, while Web services data sources provide XML files, and SAP data sources use tables.

Establish connections by supplying an underlying driver and a connection string. Together, the driver and string allow you to address the data source, and provide you a mechanism by which to set the appropriate user authentication credentials and connection properties that describe the connection instance. Once a connection is established, SAP Mobile Server can open and close it as required.

Connection Pools

SAP Mobile Server maintains database connections in a connection pool, which is a cache database connections for the cache database or any other database data source.

A connection can be reused when the database receives future requests for data, thereby improving SAP Mobile Server performance. If all the connections are being used, and the `maxPoolSize` value you configured for a connection pool has been reached, a new connection is added to the pool. For SAP Mobile Server, connection pools are based on an existing template created for a specific data source type.

See also

- *Creating and Enabling a New Domain* on page 101
- *Deleting a Domain* on page 102
- *Registering a Domain Administrator User* on page 103
- *Assigning Domain Administrators to a Domain* on page 104
- *Viewing Applications for a Domain* on page 105
- *Viewing Application Connections for a Domain* on page 105
- *Scheduling Accumulated Data Cleanup for Domains* on page 107
- *Domain Logs* on page 110

- *Client Logs* on page 162
- *Configuring Domain Security* on page 195

Connection Templates

A connection template is a model or pattern used to standardize connection properties and values for a specific connection pool type so that they can be reused. A template allows you to quickly create actual connections.

Often, setting up a connection for various enterprise data sources requires each administrator to be aware of the mandatory property names and values for connecting to data sources. Once you create a template and add appropriate property names and corresponding values (for example user, password, database name, server name, and so on), you can use the template to instantiate actual connection pools with predefined property name and value pairs.

See also

- *EIS Data Source Connection Properties Reference* on page 170
- *Creating Connections and Connection Templates* on page 168

Creating Connections and Connection Templates

Create a new connection or connection template that defines the properties needed to connect to a new data source.

1. In the left navigation pane, expand the **Domains** folder, and select the domain for which you want to create a new connection.
2. Select **Connections**.
3. In the right administration pane:
 - To create a new connection – select the **Connections** tab, and click **New**.
 - To create a new connection template – select the **Templates** tab, and click **New**.
4. Enter a unique **Connection pool name** or template name.
5. Select the **Connection pool type** or template type:
 - JDBC – choose this for most database connections.
 - Proxy - choose this if you are connecting to a proxy endpoint; for example, an Online Data Proxy data source or other proxy endpoint.
 - WEBSERVICE – choose this if you are connecting to a Web Services (SOAP or REST) data source.
 - SAP – choose this if you are connecting to an SAP (JCO) data source.
6. Select the appropriate template for the data source target from the **Use template** menu. By default, several templates are installed with SAP Mobile Platform; however, a production version of SAP Mobile Server may have a different default template list.
7. Template default properties appear, along with any predefined values. You can customize the template, if required, by performing one of:

- Editing existing property values – click the corresponding cell and change the value that appears.
- Adding new properties – click the **<ADD NEW PROPERTY>** cell in the Property column and select the required property name. You can then set values for any new properties.

Note: In a remote server environment, if you edit the sampledbs Server Name property, you must specify the remote IP number or server name. Using the value "localhost" causes cluster synchronization to fail.

8. Test the values you have configured by clicking **Test Connection**. If the test fails, either values you have configured are incorrect, or the data source target is unavailable. Evaluate both possibilities and try again. Only the SAP and JDBC connection pool types can test the connection. The Proxy and WEBSERVICE pool types cannot test the connection.
9. Click **OK** to register the connection pool.

Note: To whitelist an application using REST services, select **Connection pool type** as **Proxy**, enter the application endpoint URL (pointing to the EIS) in the **Address** field, and save the proxy properties.

The name appears in the available connection pools table on the Connections tab. Administrators can now use the connection pool to deploy packages.

See also

- *Application Connections* on page 318
- *Connection Templates* on page 168
- *EIS Data Source Connection Properties Reference* on page 170

Editing Connection Pools and Templates

Edit the properties and values assigned to connection pools and templates.

1. In the left navigation pane, expand the **Domains** folder, and select the domain for which you want to edit a connection.
2. Select **Connections**.
3. In the right administration pane:
 - To edit the properties of a connection pool, click the **Connections** tab.
 - To edit the properties of a connection pool template, click the **Templates** tab.
4. Select a connection pool or template from the list.
5. Click **Properties**.
 - a) Edit the property and value.
 - a) Click **Save** to save the changes.

Testing a Connection

Test connection properties of a data source to validate the connection values.

1. In the left navigation pane, expand the Domains folder, and select the domain for which you want to validate a connection.
2. Select **Connections**.
3. In the right administration pane, click the **Connections** tab.
4. Select the **Connection Pool Name** you want to validate.
5. Click **Properties**.
6. Click **Test Connection**.

If the connection test is not successful, see *Connection Test Errors* in the *Troubleshooting* guide.

Deleting a Connection Pool and Template

Delete a connection pool or template.

1. In the left navigation pane, expand the **Domains** folder, and select the domain for which you want to delete a connection.
2. Select **Connections**.
3. In the right administration pane:
 - To delete a connection pool, click the **Connections** tab.
 - To delete a connection pool template, click the **Templates** tab.
4. Select the connection pool or template you want to delete.
5. Click **Delete**.

EIS Data Source Connection Properties Reference

Name and configure connection properties when you create connection pools in SAP Control Center to enterprise information systems (EIS) .

See also

- *Application Connections* on page 318
- *Connection Templates* on page 168
- *Creating Connections and Connection Templates* on page 168

JDBC Properties

Configure Java Database Connectivity (JDBC) connection properties.

This list of properties can be used by all datasource types. SAP does not document native properties used only by a single driver. However, you can also use native driver properties, naming them using this syntax:

```
jdbc:<NativeConnPropName>=<Value>
```

Note: If SAP Mobile Server is connecting to a database with a JDBC driver, ensure you have copied required JAR files to correct locations. See *Installation Guide for Runtime*.

Name	Description	Supported values
After Insert	Changes the value to <code>into</code> if a database requires <code>insert into</code> rather than the abbreviated <code>into</code> .	<code>into</code>
Batch Delimiter	Sets a delimiter, for example, a semicolon, that can be used to separate multiple SQL statements within a statement batch.	<code><delimiter></code>
Blob Updater	Specifies the name of a class that can be used to update database BLOB (long binary) objects when the BLOB size is greater than <code>psMaximumBlobLength</code> .	<code><class name></code> The class must implement the <code>com.sybase.djc.sql.BlobUpdater</code> interface.
Clob Updater	Specifies the name of a class that can be used to update database CLOB (long string) objects when the CLOB size is greater than <code>psMaximumClobLength</code> .	<code><class name></code> The class must implement the <code>com.sybase.djc.sql.ClobUpdater</code> interface.
Code Set	Specifies how to represent a repertoire of characters by setting the value of <code>CS_SYB_CHARSET</code> for this datasource. Used when the data in the datasource is localized. If you do not specify the correct code set, characters may be rendered incorrectly.	<code>[server]</code> If the value is <code>server</code> , the value of the current application server's <code>defaultCodeSet</code> property is used.

Name	Description	Supported values
Commit Protocol	Specifies how SAP Mobile Server handles connections for a data-source at commit time, specifically when a single transaction requires data from multiple endpoints. If you use XA, the recovery log is stored in the tx_manager data-source, and its commit protocol must be optimistic. If tx_manager is aliased to another datasource (that is, one that is defined with the alias-For property), the commit protocol for that datasource must be optimistic. A last-resource optimization ensures full conformance with the XA specification. The commit protocol for all other datasources should be XA_2PC. Alternately, a transaction that accesses multiple datasources for which the commit protocols are optimistic is permitted.	[optimistic pessimistic XA_2PC] Choose only one of these protocols: • Optimistic – enables connections to be committed without regard for other connections enlisted in the transaction, assuming that the transaction is not marked for rollback and will successfully commit on all resources. Note: if a transaction accesses multiple data sources with commit protocol of "optimistic", atomicity is not guaranteed. • Pessimistic – specifies that you do not expect any multi-resource transactions. An exception will be thrown (and transaction rolled back) if any attempt is made to use more than one "pessimistic" data source in the same transaction. • XA_2PC – specifies use of the XA two phase commit protocol. If you are using two phase commit, then the recovery log is stored in the "tx_manager" data source, and that data source (or the one it is aliased to) must have the commit protocol of "optimistic" or "pessimistic". All other data sources for which atomicity must be ensured should have the "XA_2PC" commit protocol.

Name	Description	Supported values
Data Source Class	Sets the class that implements the JDBC datasource. Use this property (along with the driverClass property) only if you do not have a predefined database-type entry in SAP Mobile Server for the kind of SQL database you are connecting to. For example, you must use this property for MySQL database connections. You can implement a datasource class to work with a distributed transaction environment. Because SAP Mobile Server supports distributed transactions, some datasources may require that a datasource class be implemented for SAP Mobile Server to interact with it. For two-phase transactions, use the xaDataSourceClass connection property instead.	<code><com.mydata-source.jdbc.Driver></code>
Database Command Echo	Echoes a database command to both the console window and the server log file. Use this property to immediately see and record the status or outcome of database commands. When you enable this property, SAP Mobile Server echoes every SQL query to <code>ml.log</code>, which may help you debug your application.	<code>[true false]</code> Set a value of 1 to echo the database commands like <code>databaseStartCommand</code>, and <code>databaseStopCommand</code>. Otherwise, do not set this property, or use a value of 0 to disable the echo.

Name	Description	Supported values
Database Create Command	Specifies the operating system command used to create the database for this datasource. If this command is defined and the file referenced by \${databaseFile} does not exist, the command is run to create the database when an application component attempts to obtain the first connection from the connection pool for this datasource.	<command> Example: SMP_HOME\Servers\SQLAny-where11\BIN32\dbinit -q \${databaseFile}
Database File	Indicates the database file to load when connecting to a datasource. Use this property when the path to the database file differs from the one normally used by the database server. If the database you want to connect to is already running, use the databaseName connection parameter.	<string> Supply a complete path and file name. The database file you specify must be on the same host as the server.

Name	Description	Supported values
Database Name	Identifies a loaded database with which to establish a connection, when connecting to a datasource. Set a database name, so you can refer to the database by name in other property definitions for a datasource. If the database to connect to is not already running, use the database-File connection parameter so the database can be started. Note: For SAP Mobile Server, you typically do not need to use this property. Usually, when you start a database on a server, the database is assigned a name. The mechanism by which this occurs varies. An administrator can use the DBN option to set a unique name, or the server may use the base of the file name with the extension and path removed.	[DBN default] If you set this property to default, the name is obtained from the DBN option set by the database administrator. If no value is used, the database name is inherited from the database type.
Database Start Command	Specifies the operating system command used to start the database for this datasource. If this command is defined and the database is not running, the command is run to start the database when the datasource is activated.	<command> Example: SMP_HOME\Servers\SQLAny-where11\BIN32\dsrvXX.exe
Database Stop Command	Specifies the operating system command used to stop the database for this datasource. If this property is defined and the database is running, this command executes during shutdown.	<command> For a SQL Anywhere® database, where the user name and password are the defaults (dba and sql), enter: SMP_HOME\Servers\SQLAny-where11\BIN32\dsrvXX.exe
Database Type	Specifies the database type.	<database type>

Name	Description	Supported values
Database URL	Sets the JDBC URL for connecting to the database if the datasource requires an Internet connection. Typically, the server attempts to construct the database URL from the various connection properties you specify (for example, portNumber, databaseName). However, because some drivers require a special or unique URL syntax, this property allows you to override the server defaults and instead provide explicit values for this URL.	<JDBCurl> The database URL is JDBC driver vendor-specific. For details, refer to the driver vendor's JDBC documentation.
Driver Class	Sets the name of the class that implements the JDBC driver. Use this property (along with the dataSourceClass property) only if you do not have a predefined database-type entry in SAP Mobile Server for the kind of SQL database you are connecting to. For example, MySQL database connections require you to use this connection property. To create a connection to a database system, you must use the compatible JDBC driver classes. SAP does not provide these classes; you must obtain them from the database manufacturer.	<Class.forName("foo.bar.Driver")> Replace <Class.forName("foo.bar.Driver")> with the name of your driver.
Driver Debug	Enables debugging for the driver.	[true false] Set to true to enable debugging, or false to disable.
Driver Debug Settings	Configures debug settings for the driver debugger.	[default <setting>] The default is STATIC:ALL.

Name	Description	Supported values
Initial Pool Size	Sets the initial number of connections in the pool for a datasource. In general, holding a connection causes a less dramatic performance impact than creating a new connection. Keep your pool size large enough for the number of concurrent requests you have; ideally, your connection pool size should ensure that you never run out of available connections. The initialPoolSize value is applied to the next time you start SAP Mobile Server.	<int> Replace <int> with an integer to preallocate and open the specified number of connections at start-up. The default is 0. SAP suggests that you start with 0, and create additional connections as necessary. The value you choose allows you to create additional connections before client synchronization requires the server to create them.
Is Download Zipped	Specifies whether the driver file downloaded from jdbcDriverDownloadURL is in .ZIP format. This property is ignored if the value of jdbcDriverDownloadURL connection is an empty string.	[True False] The default is false. The file is copied, but not zipped to <i>SMP_HOME\lib\jdbc</i>. Set isDownloadZipped to true to save the file to <i>SMP_HOME\lib\jdbc</i> and unzip the archived copy.
JDBC Driver Download URL	Specifies the URL from which you can download a database driver. Use this property with isDownloadZipped to put the driver in an archive file before the download starts.	<URL> Replace <URL> with the URL from which the driver can be downloaded.

Name	Description	Supported values
Language	For those interfaces that support localization, this property specifies the language to use when connecting to your target database. When you specify a value for this property, SAP Mobile Server: • Allocates a CS_LOCALE structure for this connection • Sets the CS_SYB_LANG value to the language you specify • Sets the Microsoft SQL Server CS_LOC_PROP connection property with the new locale information SAP Mobile Server can access Unicode data in an Adaptive Server® 12.5 or later, or in Unicode columns in Adaptive Server 12.5 or later. SAP Mobile Server automatically converts between double-byte character set (DBCS) data and Unicode, provided that the Language and CodeSet parameters are set with DBCS values.	<language> Replace <language> with the language being used.
Max Idle Time	Specifies the number of seconds an idle connection remains in the pool before it is dropped.	<int> If the value is 0, idle connections remain in the pool until the server shuts down. The default is 60.

Name	Description	Supported values
Max Pool Size	Sets the maximum number of connections allocated to the pool for this datasource. Increase the <code>maxPoolSize</code> property value when you have a large user base. To determine whether a value is high enough, look for <code>ResourceMonitorTimeoutException</code> exceptions in <code><hostname>-server.log</code>. Continue increasing the value, until this exception no longer occurs. To further reduce the likelihood of deadlocks, configure a higher value for <code>maxWaitTime</code>. To control the range of the pool size, use this property with <code>minPoolSize</code>.	<code><int></code> A value of 0 sets no limit to the maximum connection pool size. The default is 10.
Max Wait Time	Sets the maximum number of seconds to wait for a connection before the request is cancelled.	<code><int></code> The default is 60.
Max Statements	Specifies the maximum number of JDBC prepared statements that can be cached for each connection by the JDBC driver. The value of this property is specific to each JDBC driver.	<code><int></code> A value of 0 (default) sets no limit to the maximum statements.
Min Pool Size	Sets the minimum number of connections allocated to the pool for this datasource.	<code><int></code> A value of 0 (default) sets no limit to the minimum connection pool size.

Name	Description	Supported values
Network Protocol	Sets the protocol used for network communication with the datasource. Use this property (along with the driverClass, and dataSourceClass properties) only if you do not have a predefined database-type entry in SAP Mobile Server for the kind of SQL database you are connecting to. For example, you may be required to use this property for MySQL database connections.	The network protocol is JDBC driver vendor-specific. There are no predefined values. See the driver vendor's JDBC documentation.
Password	Specifies the password for connecting to the database.	[default <password>]
Ping and Set Session Auth	Runs the ping and session-authorization commands in a single command batch; may improve performance. You can only enable the Ping and Set Session Auth property if you have enabled the Set Session Auth property so database work runs under the effective user ID of the client.	[True False] Set to true to enable, or false to disable.
Ping Connections	Pings connections before attempting to reuse them from the connection pool.	[True False] Set to true to enable ping connections, or false to disable.
Ping SQL	Specify the SQL statement to use when testing the database connection with ping.	[default <statement>] Replace <statement> with the SQL statement identifier. The default is "select 1".
Port Number	Sets the server port number where the database server listens for connection requests.	[default <port>] Replace <port> with the TCP/IP port number to use (that is, 1 – 65535). If you set the value as default, the default protocol of the datasource is used.

Name	Description	Supported values
PS Max Blob Length	Indicates the maximum number of bytes allowed when updating a BLOB datatype using Prepared-Statement.setBytes.	[default <int>] Replace <int> with the number of bytes allowed during an update. The default is 16384.
PS Max Clob Length	Indicates the maximum number of characters allowed when updating a CLOB datatype using Prepared-Statement.setString.	[default <int>] Replace <int> with the number of bytes allowed during an update. The default is 16384.
Role Name	Sets the database role that the user must have to log in to the database.	[default <name>] If you set this value to default, the default database role name of the data-source is used.
Server Name	Defines the host where the database server is running.	<name> Replace <name> with an appropriate name for the server.
Service Name	Defines the service name for the data-source. For SQL Anywhere servers, use this property to specify the database you are attaching to.	<name> Replace <name> with an appropriate name for the service.
Set Session Auth	Establishes an effective database identity that matches the current mobile application user. If you use this property, you must also use setSessionAuthSystemID to set the session ID. Alternately you can pingAndSetSessionAuth if you are using this property with pingConnection. The pingAndSetSessionAuth property runs the ping and session-authorization commands in a single command batch, which may improve performance.	[true false] Choose a value of 1 to use an ANSI SQL set session authorization command at the start of each database transaction. Set to 0 to use session-based authorizations.

Name	Description	Supported values
Set Session Auth System ID	If Set Session Authorization is enabled, specifies the database identity to use when the application server accesses the database from a transaction that runs with "system" identity.	<database identity> Replace <database identity> with the database identifier.
Start Wait	Sets the wait time (in seconds) before a connection problem is reported. If the start command completes successfully within this time period, no exceptions are reported in the server log. startWait time is used only with the databaseStartCommand property.	<int> Replace <int> with the number of seconds SAP Mobile Server waits before reporting an error.
Truncate Nanoseconds	Sets a divisor/multiplier that is used to round the nanoseconds value in a java.sql.Timestamp to a granularity that the DBMS supports.	[default <int>] The default is 10 000 000.
Use Quoted Identifiers	Specifies whether or not SQL identifiers are quoted.	[True False] Set to true to enable use of quoted identifiers, or false to disable.
User	Identifies the user who is connecting to the database.	[default <user name>] Replace <user name> with the database user name.
XA Data Source Class	Specifies the class name or library name used to support two-phase commit transactions, and the name of the XA resource library.	<class name> Replace <class name> with the class or library name. - SQL Anywhere database: com.sybase.jdbc3.jdbc.SybXADataSource - Oracle database: oracle.jdbc.xa.client.OracleXADataSource

See also

- *SAP Java Connector Properties* on page 183
- *SAP doec-abbrev PropertiesDOE-C Properties* on page 187
- *Web Services Properties* on page 190
- *Proxy Properties* on page 192

SAP Java Connector Properties

Configure SAP Java Connector (JCo) connection properties.

For a comprehensive list of SAP JCo properties you can use to create an instance of a client connection to a remote SAP system, see [http://help.sap.com/javadocs/NW04/current/jc/com/sap/mw/jco/JCO.html#createClient\(java.util.Properties\)](http://help.sap.com/javadocs/NW04/current/jc/com/sap/mw/jco/JCO.html#createClient(java.util.Properties)).

Note: SAP does not document all native endpoint properties. However, you can add native endpoint properties, naming them using this syntax:

```
<NativeConnPropName>=<SupportedValue>
```

Standard properties that can be configured within SAP Control Center include:

Table 12. General Connection Parameters for Standard JCo Properties

Name	Description	Supported Values
Enable ABAP Debugging	Enables or disables ABAP debugging. If enabled, the connection is opened in debug mode and you can step through the invoked function module in the debugger. For debugging, an SAP graphical user interface (SAPGUI) must be installed on the same machine the client program is running on. This can be either a normal Windows SAPGUI or a Java GUI on Linux/UNIX systems.	Not supported. Do not set this parameter.
Remote GUI	Specifies whether a remote SAP graphical user interface (SAPGUI) should be attached to the connection. Some older BAPIs need an SAPGUI because they try to send screen output to the client while executing.	Not supported. Do not set this parameter.

Name	Description	Supported Values
Get SSO Ticket	Generates an SSO2 ticket for the user after login to allow single sign-on. If RfcOpenConnection() succeeds, you can retrieve the ticket with RfcGet-PartnerSSOTicket() and use it for additional logins to systems supporting the same user base.	Not accessible by the customer. Do not set this parameter or leave it set to 0.
Use X509	SAP Mobile Platform sets this property when a client uses an X509 certificate as the login credential.	If an EIS RFC operation is flagged for SSO (user name and password personalization keys selected in the authentication parameters), then SAP Mobile Platform automatically sets the appropriate properties to use X.509, SSO2, or user name and password SSO credentials. The corresponding properties should not be set by the administrator on the SAP endpoint.
Additional GUI Data	Provides additional data for graphical user interface (GUI) to specify the SAP router connection data for the SAPGUI when it is used with RFC.	Not supported.
GUI Redirect Host	Identifies which host to redirect the remote graphical user interface to.	Not supported.
GUI Redirect Service	Identifies which service to redirect the remote graphical user interface to.	Not supported.
Remote GUI Start Program	Indicates the program ID of the server that starts the remote graphical user interface.	Not supported.

Properties that can be configured manually within SAP Control Center include:

Table 13. General Connection Parameters for Manual JCo Properties

Name	Description	Supported Values
jco.client.cpic_trace	Enables and disables CPIC trace	-1 - take over environment value <CPIC_TRACE>] 0 - no trace 1, 2, 3 - different trace levels
jco.client.delta	Enables and disables table parameter delta management	1 - enable (default) 0 - disable
jco.client.deny_initial_password	Deny usage of initial passwords	0 - default 1
jco.client.extid_data	External identification user login data	
jco.client.extid_type	Type of external identification user login data	
jco.client.msserv	SAP message server port to use instead of the default sapms<sysid>	
jco.client.saprouter	SAP router string to use for a system protected by a firewall	
jco.client.snc_sso	Turns on or off SSO of SNC mechanism	1 - yes (default) 0 - no If set to 0, use user and password credentials instead
jco.destination.auth_type	Authentication type	Configured user or current user
jco.destination.expiration_check_period	Interval, in milliseconds, with which the timeout checker thread checks the connections in the pool for expiration	
jco.destination.expiration_time	Time, in milliseconds, after which the connections held by the internal pool may be closed	

Name	Description	Supported Values
jco.destination.max_get_client_time	Maximum time, in milliseconds, to wait for a connection, if the maximum allowed number of connections is allocated by the application	
jco.destination.one_roundtrip_repository	If the property is not set, the destination makes a remote call to check if RFC_METADATA_GET is available, and when available uses it	• 1 - forces the usage of RFC_METADATA_GET in the ABAP system • 0 - deactivates RFC_METADATA_GET in the ABAP system
jco.destination.peak_limit	Maximum number of active connections that can be created for a destination simultaneously	
jco.destination.pool_capacity	Maximum number of idle connections kept open by the destination	A value of 0 sets no connection pooling, so connections are closed after each request
jco.destination.repository.passwd	The password for a repository user	Mandatory if a repository user is used
jco.destination.repository.snc_mode	(Optional) If SNC is used for this destination, it is possible to turn it off for repository connections, if this property is set to 0	Defaults to the value of jco.client.snc_mode
jco.destination.repository.user	(Optional) If the repository destination is not set, and this property is set, it acts as a user for repository queries, enabling a different user for repository lookups and restriction of permissions accordingly	
jco.destination.repository_destination	Specifies which destination should be used for repository queries	
jco.destination.user_id	Login user that identifies the user when not using user and password for login	The parameter is only required if neither user nor user alias is provided

See also

- *JDBC Properties* on page 171
- *SAP doec-abbrev PropertiesDOE-C Properties* on page 187
- *Web Services Properties* on page 190
- *Proxy Properties* on page 192

SAP doec-abbrev PropertiesDOE-C Properties

Configure SAP® Data Orchestration Engine Connector (DOE-C) properties. This type of connection is available in the list of connection templates only when you deploy a SAP® Data Orchestration Engine Connector package. No template exists for these types of connections.

Note: If you change the username or password property of a DOE-C connection, you must reopen the same dialog and click **Test Connection** after saving. Otherwise the error state of this DOE-C package is not set properly, and an error message is displayed. This will not work if you click **Test Connection** before saving the properties.

Name	Description	Supported values
Username	Specifies the SAP user account ID. The SAP user account is used during interaction between the connected SAP system and client for certain administrative activities, such as sending acknowledgment messages during day-to-day operations or "unsubscribe" messages if a subscription for this connection is removed. This account is not used for messages containing business data; those types of messages are always sent within the context of a session authenticated with credentials provided by the mobile client. The technical user name and password or certificateAlias must be set to perform actions on subscriptions. The certificateAlias is mutually exclusive with and overrides the technical user name and password fields if set. The technical user name and password fields can be empty, but only if certificateAlias is set.	Valid SAP login name for the DOE host system.

Name	Description	Supported values
Password	Specifies the password for the SAP user account.	Valid password.
DOE SOAP Timeout	Specifies a timeout window during which unresponsive DOE requests are aborted.	Positive value (in seconds). The default is 420 (7 minutes).
DOE Extract Window	Specifies the number of messages allowed in the DOE extract window.	Positive value (in messages). The minimum value is 10. The maximum value is 2000. The default is 50. When the number of messages in the DOE extract window reaches 50% of this value, DOE-C sends a <code>Status-ReqFromClient</code> message, to advise the SAP DOE system of the client's messaging status and acknowledge the server's state.
Packet Drop Size	Specifies the size, in bytes, of the largest JavaScript Object Notation (JSON) message that the DOE connector processes on behalf of a JSON client. The packet drop threshold size should be carefully chosen, so that it is larger than the largest message sent from the DOE to the client, but smaller than the maximum message size which may be processed by the client.	Positive value (in bytes). The default is 1048576 bytes (1MB). Do not set lower than 4096 bytes; there is no maximum limitation.

Name	Description	Supported values
Service Address	Specifies the DOE URL.	Valid DOE URL. If you are using DOE-C with SSL: • Modify the port from the standard <code>http://host:8000</code> to <code>https://host:8001/</code>. • Add the certificate being used as the technical user and DOE-C endpoint security profile certificate to the SAP DOE system's SSL Server certificate list by using the STRUST transaction. See your SAP documentation for details.
Listener URL	Specifies the DOE-C server listener URL.	Valid DOE-C listener URL, for example <code>http://<sup_host-name>:8000/doi/publish</code> .
Certificate Alias	Sets the alias for the SAP Mobile Platform keystore entry that contains the X.509 certificate for SAP Mobile Server's SSL peer identity. If you do not set a value, mutual authentication for SSL is not used when connecting to the Web service. If you are using DOE-C with SSO use the "SAP Technical User Certificate Alias" only for configurations which require the technical user to identify itself using an X.509 certificate; it specifies the Certificate Alias to be used as the technical user. This overrides the "Username" and "Password" settings normally used.	Valid certificate alias.

See also

- *JDBC Properties* on page 171
- *SAP Java Connector Properties* on page 183
- *Web Services Properties* on page 190
- *Proxy Properties* on page 192

Web Services Properties

Configure connection properties for the Simple Object Access Protocol (SOAP) and Representational State Transfer (REST) architectures.

Name	Description	Supported Values
Password	Specifies the password for HTTP basic authentication, if applicable.	Password
Address	Specifies a different URL than the port address indicated in the WSDL document at design time.	HTTP URL address of the Web service. A backslash is appended to the address URL, if it does not already exist in the URL you specify.
User	Specifies the user name for HTTP basic authentication, if applicable.	User name
Certificate Alias	Sets the alias for the SAP Mobile Platform keystore entry that contains the X.509 certificate for SAP Mobile Server's SSL peer identity. If you do not set a value, mutual authentication for SSL is not used when connecting to the Web service.	Use the alias of a certificate stored in the SAP Mobile Server certificate keystore.
Preemptive Authentication	When credentials are available and this property is set to the default of false, it allows SAP Mobile Server to send the authentication credentials only in response to the receipt of a server message in which the HTTP status is 401 (UNAUTHORIZED) and the WWW-Authenticate header is set. In this case, the message exchange pattern is: request, UNAUTHORIZED response, request with credentials, service response. When set to true and basic credentials are available, this property allows SAP Mobile Server to send the authentication credentials in the original SOAP or REST HTTP request message. The message exchange pattern is: request with credentials, a service response.	False (default) True

Name	Description	Supported Values
Socket Timeout	The socket timeout value controls the maximum time, in milliseconds, after a Web service operation (REST or SOAP) is allowed to wait for a response from the remote system; if the EIS does not respond in that time, the operation fails and the SMP thread is unblocked.	Time in milliseconds (default: 60000). Range of [0 – 2147483647], where 0 is interpreted as infinity.
credential.<X>.name	Defines the EIS connection definition to identify the NamedCredential. For more information on token-based SSO using NamedCredential, see <i>Single Sign-on Using NamedCredential in Security</i> .	- credential.<X>.name=credential name - header:<HTTP header name> - cookie: <HTTP cookie name>
credential.<X>.mapping	Defines how the NamedCredential should be propagated to the EIS. For more information on token-based SSO using NamedCredential, see <i>Single Sign-on Using NamedCredential</i> .	credential.<X>.mapping=credential mapping to header/cookie
http.header.X.set	Specifies the name of the static header to add to the Web service request (SOAP or REST).	Header name
http.header.X.value	Specifies the content of the header to add to the Web service request (SOAP or REST).	Header value
http.cookie.X.set	Specifies the name of the static cookie to add to the Web service request (SOAP or REST).	Cookie name
http.cookie.X.value	Specifies the content of the cookie to add to the Web service request (SOAP or REST).	Cookie value

See also

- *JDBC Properties* on page 171
- *SAP Java Connector Properties* on page 183
- *SAP doec-abbrev Properties**DOE-C Properties* on page 187

- *Proxy Properties* on page 192

Proxy Properties

(Applies only to OData SDK Application and REST API applications). Proxy properties identify the application endpoint and the pool size for connections to a Proxy server.

Name	Description	Supported Values
Allow Anonymous Access	While using REST services, you can enable/disable anonymous user access. Whenever you create multiple proxy connections in the connection pool with the same proxy endpoint, set Allow Anonymous Access to True in all of those proxy connections. This configuration ensures that the endpoint is whitelisted only once in the connection pool.	The default value is False. To enable anonymous user access, set the value to True.
Certificate Alias	Sets the alias for the SAP Mobile Platform keystore entry that contains the X.509 certificate for SAP Mobile Server's SSL peer identity.	Use the alias of a certificate stored in the SAP Mobile Server certificate keystore.
Address	Corresponds to the application endpoint provided when registering an application.	Must be a valid application endpoint.
Pool Size	Determines the maximum number of connections allocated to the pool for this datasource.	The default value set for the pool size is 25. However, if Netweaver Gateways is used as the HTTP Proxy, ensure that enough work processes exist in both the Gateway system and in any SAP EIS systems (for example, SAP ERP or CRM) to handle the peak load. Generally, SAP recommends that you limit this value to be less than 70% of back-end concurrent working processes for the Gateway. For example, if the Gateway uses the maximum of 25, then set this value to 10 or 12.

Name	Description	Supported Values
Enable URL Rewrite	Enables the proxy component to rewrite URLs (or URIs) embedded in a response, with SAP Mobile Platform URLs that cause the client requests to be directed back to the SAP Mobile Server (proxy component), rather than to the back end server.	The default value is True. To disable URL rewrite, set the value to False.
Enable Http Proxy	Sets the connection to an EIS using an HTTP proxy. If you set this property to True, you must configure an HTTP proxy host and port on the server. For more information, see <i>Configuring SAP Mobile Server to Securely Communicate With an HTTP Proxy in SAP Control Center for SAP Mobile Platform</i> .	The default value is set to False.
Password	Corresponds to the password of the backend system.	
Socket Timeout	Proxy connection timeout, in milliseconds.	The default value comes from the global settings in the cluster configuration: Cluster > Configuration > Performance > Connection Global > click Properties > Update 'Proxy connection timeout(s)' > click OK > click Save . If you enter a Socket Timeout value in the Proxy Connection profile, that value takes precedence.
User	Corresponds to the username of the backend system.	

Name	Description	Supported Values
URL Rewrite Prefixes	Include URL prefixes (and defined sub-URLs) embedded in a response, to be rewritten by the proxy component. You can specify multiple URL prefixes using comma separated strings. For example, /catalog1/subcatalog1, /catalog3. EnableURLRewrite proxy property must be true. The proxy, before returning the response to the client, performs URL rewrite. This property identifies the relative paths present in the back end response, and rewrites them appropriately. when a response comes from a back end, the proxy looks for the defined prefixes in the response body and prefixes those relative URLs.	URL prefixes are case sensitive and must not have spaces between them.

Note:

- In SAP Control Center, when the application endpoint for a registered application is modified under the **Applications** node, you must manually update the **Address** in the proxy properties of the connection pool.
- In SAP Control Center, in addition to the application endpoint, you must register any URL that is required by an application for a proxy service to enable communication with SAP Mobile Server.
- The application end point should be white-listed only once as a proxy connection. The proxy connection name should be same as the application ID, if an application is registered to be used for referencing the EIS service end point.

See also

- *JDBC Properties* on page 171
- *SAP Java Connector Properties* on page 183
- *SAP doec-abbrev Properties* *DOE-C Properties* on page 187
- *Web Services Properties* on page 190
- *Configuring Secure Communication With an HTTP Proxy* on page 44

Configuring Domain Security

Configure security for an individual domain to meet the customer's security requirements.

Prerequisites

Before mapping and assigning administrator roles, ensure that you have set the SAP Mobile Platform administration and user roles and passwords required for SAP Control Center administrator login. See *Enabling Authentication and RBAC for Administrator Logins* in *Security*.

Task

Perform steps to appropriately configure domain security settings.

1. *Choosing a Security Configuration*

Select a security configuration that provides authentication and optionally authorization, attribution, or auditing services. You can assign as many security configurations as needed to a domain.

2. *Assigning Domain Administrators to a Domain*

Assign domain administration privileges to a domain administrator. You must be a platform administrator to assign and unassign domain administrators.

3. *Mapping Roles for a Domain*

Map logical roles to physical roles for a domain by setting the mapping state. Domain administrators map roles for the domains they control. Role mappings performed at the domain level are automatically applied to all domains that share the same security configuration. Domain-level role mapping overrides mapping set at the cluster level by the platform administrator.

See also

- *Creating and Enabling a New Domain* on page 101
- *Deleting a Domain* on page 102
- *Registering a Domain Administrator User* on page 103
- *Assigning Domain Administrators to a Domain* on page 104
- *Viewing Applications for a Domain* on page 105
- *Viewing Application Connections for a Domain* on page 105
- *Scheduling Accumulated Data Cleanup for Domains* on page 107
- *Domain Logs* on page 110
- *Client Logs* on page 162
- *Connections* on page 167

Choosing a Security Configuration

Select a security configuration that provides authentication and optionally authorization, attribution, or auditing services. You can assign as many security configurations as needed to a domain.

Only super administrators have privileges to create security configurations. Domain administrators can view a security configuration only after a super administrator has assigned it to the domain.

1. In the left navigation pane, navigate to **Cluster > Domains > <DomainName> > Security**.
2. In the right administration pane, select the **General** tab and click **Assign**. The **Assign Security Configurations** dialog appears.
3. Select one or more security configurations to assign to the domain by checking the box adjacent to the configuration name.
4. Click **OK**.
A message appears above the right administration pane menu indicating the success or failure of the assignment. If successful, the new security configuration appears in the list of security configurations.
5. To set the default security configuration for the domain, check the box adjacent to the configuration name and click **Set Default**.
6. To remove a security configuration, check the box adjacent to the configuration name and click **Unassign**.

You cannot remove a security configuration if:

- it is mapped to one or more MBO packages
- it is the default security configuration for the domain
- it is used by an application connection template.

Assigning Domain Administrators to a Domain

Assign domain administration privileges to a domain administrator. You must be a platform administrator to assign and unassign domain administrators.

Prerequisites

Ensure the user is already registered as a domain administrator in the Domain Administrators tab.

Task

1. In the left navigation pane, expand the **Domains** folder, and select the domain for which to assign domain administration privileges.
2. Select the domain-level **Security** folder.

3. In the right administration pane, select the **Domain Administrators** tab, and click **Assign**.
4. Select one or more administrator users to assign to the domain by checking the box adjacent to the user name.
5. Click **OK**.
A message appears above the right administration pane menu indicating the success or failure of the assignment. If successful, the new domain administrator appears in the list of users.

Mapping Roles for a Domain

Map logical roles to physical roles for a domain by setting the mapping state. Domain administrators map roles for the domains they control. Role mappings performed at the domain level are automatically applied to all domains that share the same security configuration. Domain-level role mapping overrides mapping set at the cluster level by the platform administrator.

Prerequisites

SAP Mobile Platform cannot query all enterprise security servers directly; to perform authentication successfully know the physical roles that are required.

Task

1. In the left navigation pane of SAP Control Center, expand **Domains** > **Domain name** > **Security** and select the security configuration to map roles for.
2. In the right administration pane, click the **Role Mappings** tab.
3. Select a logical role and select one of the following in the adjacent list:

State	Description
AUTO	To map the logical role to a physical role of the same name.
NONE	To disable the logical role, which means that the logical role is not authorized.
MAP	To manually map the logical role when the physical and logical role names do not match. See <i>Mapping a Physical Role Manually</i> .

Mapping a Physical Role Manually

Use the Role Mappings dialog to manually map required physical roles for a logical role when physical and logical role names do not match. If names do not match, the AUTO mapping state does not work.

Prerequisites

SAP Mobile Platform cannot query all supported enterprise security servers directly; for successful authentication, you must know the physical roles your back-end systems require.

Task

You can map a logical role to one or more physical roles. You can also map multiple logical roles to the same physical role. If a role does not exist, you can also add or delete names as needed.

1. Review the list of existing physical role names that you can map to the logical role you have selected. If the list retrieved is too long to locate the name quickly, either:
 - Click the banner of Available Roles list to sort names alphanumerically.
 - Start typing characters in the box, then click the Search button to filter the available list.
2. If a role that you require still does not appear, enter the **Role name** and click the + button. The role name appears in the **Available roles** list with an asterisk (*). This asterisk indicates that an available role was added by an administrator, not a developer.
3. To remove a role you no longer require from the **Available roles** list, select the name and click the x button adjacent to the **Role name** field.
The role is removed and can no longer be mapped to a logical role.
4. To map a logical role that appears in the text area of the Role Mappings dialog to a physical role:
 - a) Select one or more **Available roles**.
 - b) Click **Add**.
5. To unmap a role:
 - a) Select one or more **Mapped roles**.
 - b) Click **Remove**.
The roles are returned to the **Available roles** list.
6. Click **OK** to save these changes.

Once a logical role has been manually mapped, the mapping state changes to MAPPED. The roles you have mapped appear in the active Physical Roles cell for either a package-specific or server-wide role mappings table.

Mapping State Reference

The mapping state determines the authorization behavior for a logical name instance.

State	Description
AUTO	Map the logical role to a physical role of the same name. The logical role and the physical role must match, otherwise, authorization fails.
NONE	Disable the logical role, which means that the logical role is not authorized. This mapping state prohibits anyone from accessing the resource (MBO or Operation). Carefully consider potential consequences before using this option.

State	Description
MAPPED	A state that is applied after you have actively mapped the logical role to one or more physical roles. Click the cell adjacent to the logical role name and scroll to the bottom of the list to see the list of mapped physical roles.

Security Configurations

SAP Mobile Platform does not provide proprietary security systems for storing and maintaining users and access control rules, but delegates these functions to the enterprise's existing security solutions.

A security configuration determines the scope of user identity, performs authentication and authorization checks, and can be assigned multiple levels (domain or package). Applications inherit a security configuration when the administrator assigns the application to a domain via a connection template.

Users can be authenticated differently, depending on which security configuration is used. For example, a user identified as "John" may be authenticated different ways, depending on the named security configuration protecting the resource he is accessing: it could be an MBO package, a DCN request, use of SAP Control Center.

The anonymous security configuration provides unauthenticated user access, and is targeted to applications that do not require tight security.

The Agency security configuration provides pass-through authentication to the Agency Server for Agency applications. The Agency security configuration employs the NoSecLoginModule to allow user credentials to be sent to the Agency server for authentication. SAP Mobile Platform does not authenticate this security configuration.

Security configurations aggregate various security mechanisms for protecting SAP Mobile Platform resources under a specific name, which administrators can then assign. Each security configuration consists of:

- A set of configured security providers. Security provider plug-ins for many common security solutions are included with the SAP Mobile Platform.
- Role mappings (which are set at the domain and package level) that map logical roles to back end physical roles.

A user entry must be stored in the security repository used by the configured security provider to access any resources (that is, either a SAP Control Center administration feature or an application package that accesses data sets from a back-end data source). When a user attempts to access a particular resource, SAP Mobile Server tries to authenticate and authorize the user, by checking the security repository for:

- Security access policies on the requested resource

- Role memberships

1. *Creating a Security Configuration*

Create and name a set of security providers and physical security roles to protect SAP Mobile Platform resources.

2. *Assigning a Security Configuration to a Domain*

Assign security configurations to one or more domains. This allows the supAdmin to offer a security repository for application user authentication and authorization, as well as to share security providers across domains in case one tenant uses multiple domains.

3. *Viewing Security Configuration Usage*

You can view the security configurations and logical roles used to access applications. This enables you to assess the impact of changing a security configuration or logical role.

4. *Anonymous Access Security Configuration*

Allow unauthenticated users access to application data, for example, applications that allow users to browse a read-only product catalog without logging in by assigning the anonymous security configuration to the application.

5. *Agency Security Configuration*

The Agency security configuration allows Agency application users to be authenticated by the Agency Server.

6. *SiteMinder Authentication with SAP Mobile Platform*

Configure your SiteMinder environment for authentication in SAP Mobile Platform.

See also

- *Configuring Domain Security* on page 195
- *Creating Logical Roles for a Security Configuration* on page 241
- *Mapping a Physical Role Manually* on page 243

Creating a Security Configuration

Create and name a set of security providers and physical security roles to protect SAP Mobile Platform resources.

Only platform administrators can create security configurations. Domain administrators can only view after the platform administrator creates and assigns them to a domain.

1. In the left navigation pane of SAP Control Center, select **Security**.
2. In the right administration pane, select the **General** tab, and click **New**.
3. Enter a name for the security configuration and click **OK**.
4. In the left navigation pane, under **Security**, select the new security configuration.
5. In the right administration pane, select the Settings tab, and set values for these properties as required:

- **Authentication cache timeout** – determines how long authentication results should be cached before a user is required to reauthenticate. For details, see *Authentication Cache Timeouts* in *Security*. Set the cache timeout value in seconds. The default is 3600. To force re-authentication, change this value to 0.
- **Maximum number of failed authentications** – determines the maximum number of login attempts after which the user is locked.
- **Authentication lock duration** – determines how long the user is locked after the maximum login attempts is reached.

6. Click **Save**.

7. Select the tab corresponding to the type of security provider you want to configure: Authentication, Authorization, Attribution or Audit.

8. To edit the properties of a preexisting security provider in the configuration:

- a) Select the provider, and click **Properties**.
- b) Configure the properties associated with the provider by setting values according to your security requirements. Add properties as required. For more information about configuring security provider properties, see the individual reference topics for each provider.
- c) Click **Save**.

9. To add a new security provider to the configuration:

- a) Click **New**.
- b) Select the provider you want to add.
- c) Configure the properties associated with the provider by setting values according to your security requirements. Add properties as required. For more information about configuring security provider properties, see the individual reference topics for each provider.
- d) Click **OK**.

The configuration is saved locally, but not yet committed to the server.

Note: When you create a new security configuration, SAP Mobile Platform sets the NoSecurity provider by default. SAP recommends that after you add, configure, and validate your providers, you remove the NoSecurity provider. For more information on the NoSecurity provider, see *NoSecurity Configuration Properties*.

10. To remove the NoSecurity provider:

- a) In the left navigation pane, expand **Security** folder and select the new security configuration.
- b) In the right administration pane, within the **Authentication** tab, select **NoSecLoginModule** and click **Delete**.
- c) In the right administration pane, within the **Authorization** tab, select **NoSecAuthorizer** and click **Delete**.
- d) In the right administration pane, within the **Attribution** tab, select **NoSecAttributer** and click **Delete**.

- e) In the **General** tab, click **Apply**.
- 11. To map roles for the security configuration, select the **Role Mappings** tab. Create logical roles and map them to physical roles in the security provider.
- 12. Select the **General** tab, and click **Validate** to confirm that SAP Mobile Server accepts the new security configuration.
A message indicating the success of the validation appears above the menu bar.
- 13. Click **Apply** to save changes to the security configuration, and apply them across SAP Mobile Server.
A message indicating the success of the application appears above the menu bar.

See also

- *Assigning a Security Configuration to a Domain* on page 246

Security Providers

Configure one or more security providers as part of a named security configuration. There are different types of providers you can use, and these can be ordered and flagged according to the requirements of your production environment.

Configure security providers for SAP Mobile Server by logging in to the server in SAP Control Center and clicking **Security > Security Configuration**.

For third-party providers or providers you have created with the CSI SDK, save related JAR files or DLLs in the `SMP_HOME\Servers\UnwiredServer\lib\ext` folder.

- Authentication modules – verify the identity of a user accessing a network with the mobile application, typically via a login form or some other login or validation mechanism. Authentication in SAP Mobile Server is distinct from authorization. You must have at least one authentication module configured in a production deployment of SAP Mobile Server. You can stack multiple providers so users are authenticated in a particular sequence.
- Authorization modules – check the access privileges for an authenticated identity. SAP recommends that you have at least one authorization module configured in a production deployment of SAP Mobile Server.
- Attribution modules – when a user is authenticated, a custom attribution provider can add more information about the authenticated user. Attribution modules are only available if you have created a custom provider with the CSI SDK and saved to the correct folder.
- Auditing modules – report all audit events to allow you to evaluate the security system implementation for SAP Mobile Server. Auditing provides you a record of all the security decisions that have been made. Each successful authentication creates a session key that shows up in subsequent security checks for that user. Unsuccessful authentications are also logged. Each authorization records what roles were checked, or what resource was accessed. Audit filters determine what events get recorded, the audit format determines what the audit records look like, and the audit destination specifies where audit records are sent. Use the audit trail to identify who did what and when, with respect to objects secured by your providers. Auditing modules are optional.

In most cases, each security module requires a unique set of configuration properties. However, there are some cases when modules require a common set of properties, and these properties are configured once for each module on a tab created for that purpose.

You can configure different security providers for administrator authentication and device user authentication. For more information on configuring security providers depending on the type of user, see either *Enabling Authentication and RBAC for User Logins* or *Enabling Authentication and RBAC for Administrator Logins* in the *Security* guide.

Stacking Providers and Combining Authentication Results

Optionally, implement multiple login modules to provide a security solution that meets complex security requirements. SAP recommends provider stacking as a means of eliciting more precise results, especially for production environment that require different authentications schemes for administrators, DCN, SSO, and so on.

Stacking is implemented with a controlFlag attribute that controls overall behavior when you enable multiple providers. Set the controlFlag on a specific provider to refine how results are processed.

For example, say your administrative users (supAdmin in a default installation) are not also users in an EIS system like SAP. However, if they are authenticated with just the default security configuration, they cannot also authenticate to the HttpAuthenticationLoginModule used for SSO2Token retrieval. In this case, you would stack a second login modules with a controlFlag=sufficient login module for your administrative users.

Or, in a custom security configuration (recommended), you may also find that you are using a technical user for DCN who is also not an SAP user. This technical user does not need SSO because they will not need to access data. However, the technical user still needs to be authenticated by SAP Mobile Server. In this case, you can also stack another login module so this DCN user can login.

1. Use SAP Control Center to create a security configuration and add multiple providers as required for authentication.
2. Order multiple providers by selecting a login module and using the up or down arrows at to place the provider correctly in the list.

The order of the list determines the order in which authentication results are evaluated.

3. For each provider:
 - a) Select the provider name.
 - b) Click **Properties**.
 - c) Configure the controlFlag property with one of the available values: required, requisite, sufficient, optional.

See *controlFlag Attribute Values* for descriptions of each available value.

- d) Configure any other common security properties as required.
4. Click **Save**.

5. Select the **General** tab, and click **Apply**.

For example, say you have sorted these login modules in this order and used these controlFlag values:

- LDAP (required)
- NT Login (sufficient)
- SSO Token (requisite)
- Certificate (optional)

The results are processed as indicated in this table:

Pro- vider	Authentication Status							
LDAP	pass	pass	pass	pass	fail	fail	fail	fail
NT Log- in	pass	fail	fail	fail	pass	fail	fail	fail
SSO To- ken	*	pass	pass	fail	*	pass	pass	fail
Certifi- cate	*	pass	fail	*	*	pass	fail	*
Overall result	pass	pass	pass	fail	fail	fail	fail	fail

Stacking LoginModules in SSO Configurations

(Not applicable to Online Data Proxy) Use LoginModule stacking to enable role-based authorization for MBOs and data change notifications (DCNs).

controlFlag Attribute Values

(Not applicable to Online Data Proxy) The SAP implementation uses the same control flag (controlFlag) attribute values and definitions as those defined in the JAAS specification.

If you stack multiple providers, you must set the control flag attribute for each enabled provider.

Control Flag Value	Description
Required	The LoginModule is required. Authentication proceeds down the Log-inModule list.

Control Flag Value	Description
Requisite	The LoginModule is required. Subsequent behavior depends on the authentication result: • If authentication succeeds, authentication continues down the LoginModule list. • If authentication fails, control returns immediately to the application (authentication does not proceed down the LoginModule list).
Sufficient	The LoginModule is not required. Subsequent behavior depends on the authentication result: • If authentication succeeds, control returns immediately to the application (authentication does not proceed down the LoginModule list). • If authentication fails, authentication continues down the LoginModule list.
Optional (default)	The LoginModule is not required. Regardless of success or failure, authentication proceeds down the LoginModule list.

Example

Providers are listed in this order and with these controlFlag:

1. CertificateAuthenticationLoginModule (sufficient)
2. LDAP (optional)
3. NativeOS (sufficient)

A client doing certificate authentication (for example, X.509 SSO to SAP) can authenticate immediately. Subsequent modules are not called, because they are not required. If there are regular user name and password credentials, they go to LDAP, which may authenticate them, and set them up with roles from the LDAP groups they belong to. Then NativeOS is invoked, and if that succeeds, SAP Mobile Platform picks up roles based on the Windows groups they are in.

LDAP Provider Stacking and Configuration Sharing

LDAP login and attribution modules can sometimes share a common configuration. LDAPAttributer can share the configuration properties from the configured LDAP login modules only if no configuration properties are explicitly configured for LDAPAttributer.

When stacking these modules, be aware that authorizers do not inherit configuration properties from the login modules you configure. Configurations must be explicit. In the case where both LDAPLoginModule and LDAPAuthorizer are separately configured in a :

- Matching configuration, then LDAPAuthorizer simply skips the role retrieval.

- Differing configuration, then LDAPAuthorizer proceeds with the role retrieval from the configured back-end, and performs the authorization checks using the complete list of roles (from both the login module and itself).

Only one attributer instance needs to be configured even when multiple login module instances are present in the security configuration. The LDAPAttributer attributes an authenticated subject using the LDAP configuration that was used to authenticate the subject. However, the list of available roles is computed by the LDAPAttributer by iterating through all available LDAP configurations.

When using LDAPAttributer stacking and configuration, keep in mind:

- LDAPAttributer has maximum functionality when combined with the LDAP authentication provider; the LDAPAttributer can be configured completely standalone or with alternate authentication providers.
- If you do not configure an LDAPLoginModule, you must define the configure all properties in the attributer.
- If explicit configuration properties are specified for the attributer, then the properties from the login module are not used for attributer functionality, including retrieving attributes for authenticated subjects, listing roles, and more. SAP recommends that you share configurations rather than trying to maintain separate configurations.

Configuring an LDAP Provider to use SSL

If your LDAP server uses a secure connection, and its SSL certificate is signed by a nonstandard certificate authority, for example it is self-signed, use SAP Control Center to import the certificate into the key store.

1. Use SAP Control Center to import the certificate into the key store. See *Managing Certificates* in *SAP Control Center for SAP Mobile Platform* for procedures to import a certificate.
2. Restart SAP Mobile Platform services.
3. Log in to SAP Control Center for SAP Mobile Platform.
4. In the navigation pane of SAP Control Center, expand the Security folder and select the desired security configuration in which to add the LDAP provider.
5. In the administration pane, click the **Authentication** tab.
6. Add an LDAPLoginModule, configuring the ProviderURL, Security Protocol, ServerType, Bind DN, Bind Password, Search Base, and other properties determined by you and the LDAP administrator. Choose **one** of the two methods below to secure a connection to the LDAP server:
 - a) Use `ldaps://` instead of `ldap://` in the **ProviderURL**.
 - b) Use `ssl` in the **Security Protocol**.
7. In the **General** tab, select **Validate** then **Apply**.
8. Click **OK**.

Reordering Configured Providers

List stacked security providers for a security configuration to identify them as primary or auxiliary providers. Providers are used in the order configured.

1. In the left navigation pane, expand the **Security** folder.
2. Select the security configuration you want to administer.
3. In the right administration pane, select the tab corresponding to the type of security provider you want to configure: Authentication, Authorization, Attribution, or Audit.
4. Select a provider from the list, then use the up and down arrows to the right of the table to achieve the desired placement.
5. Select the **General** tab, and click **Apply**.

A notification message appears if a server restart is required for changes to take effect.

Security Provider Configuration Properties

Security providers implement different properties, depending on the type of provider you are configuring.

Platform administrators can configure application security properties in the SAP Control Center. These properties are then transcribed to an XML file in the `SMP_HOME\Servers\UnwiredServer\Repository\CSI\` directory. A new section is created for each provider you add.

LDAP Configuration Properties

Use these properties to configure the LDAP provider used to authenticate SAP Control Center administration logins or to configure the LDAP provider used to authenticate device application logins. If you are configuring an LDAP provider for device application logins in SAP Control Center, then SAP Mobile Platform administrators use SAP Control Center these properties are saved to the `SMP_HOME\Servers\UnwiredServer\Repository\CSI\<security configuration name>.file`.

The Java LDAP provider consists of three provider modules.

- The **LDAPLoginModule** provides authentication services. Through appropriate configuration, you can enable certificate authentication in **LDAPLoginModule**.
- (Optional) **LDAPAuthorizer** or **RoleCheckAuthorizer** provide authorization service in conjunction with **LDAPLoginModule**. **LDAPLoginModule** works with either authorizer. The **RoleCheckAuthorizer** is part of every security configuration but does not appear in SAP Control Center.

Use **LDAPAuthorizer** only when **LDAPLoginModule** is not used to perform authentication, but roles are still required to perform authorization checks against the LDAP data store. If you use **LDAPAuthorizer**, always explicitly configure properties; for it cannot share the configuration options specified for the **LDAPLoginModule**.

- (Optional) **LDAPAttributer** is used to retrieve the list of roles from the LDAP repository. These roles are displayed in the role mapping screen in SAP Control Center. The LDAP attributer is capable of sharing the configuration properties from the LDAPLoginModules. If no configuration properties are explicitly specified, then the attributer iterates through the configured LDAPLoginModules and retrieves the roles from all the LDAP repositories configured for the different LDAPLoginModules.

Use this table to help you configure properties for one or more of the supported LDAP providers. When configuring modules or general server properties in SAP Control Center, note that properties and values can vary, depending on which module or server type you configure.

Note: The following characters have special meaning when they appear in a name in LDAP: , (comma), = (equals), + (plus), < (less than), > (greater than), # (number or hash sign), ; (semicolon), \ (backslash), / (forward slash), LF (line feed), CR (carriage return), " (double quotation mark), ' (single quotation mark), * (asterisk), ? (question mark), & (ampersand), and a space at the beginning or end of a string. LDAP providers do not handle these special characters in any of the names or DN's, in any of the configuration properties. Additionally, some of the properties, as identified below, cannot use these special characters in common names.

Property	Default Value	Description
ServerType	None	Optional. The type of LDAP server you are connecting to: • <code>sunone5</code> -- SunOne 5.x OR iPlanet 5.x • <code>msad2k</code> -- Microsoft Active Directory, Windows 2000 • <code>nsds4</code> -- Netscape Directory Server 4.x • <code>openldap</code> -- OpenLDAP Directory Server 2.x The value you choose establishes default values for these other authentication properties: • RoleFilter • UserRoleMembership • RoleMemberAttributes • AuthenticationFilter • DigestMD5Authentication • UseUserAccountControl

Property	Default Value	Description
ProviderURL	ldap://localhost:389	The URL used to connect to the LDAP server. Without this URL configured, SAP Mobile Server cannot contact your server. Use the default value if the server is: - Located on the same machine as your product that is enabled with the common security infrastructure. - Configured to use the default port (389). Otherwise, use this syntax for setting the value: ldap://<hostname>:<port>
DefaultSearchBase	None	The LDAP search base that is used if no other search base is specified for authentication, roles, attribution and self registration: - dc=<domainname>,dc=<tld> For example, a machine in sybase.com domain would have a search base of dc=sybase,dc=com. - o=<company name>,c=<country code> For example, this might be o=SAP,c=us for a machine within the SAP organization. Note: When you configure this property in the "admin" security configuration used to authenticate the administrator in SAP Control Center, the property value should not contain any special characters, as listed above, in any of the common names or distinguished names.
SecurityProtocol	None	The protocol to be used when connecting to the LDAP server. The specified value overrides the environment property java.naming.security.protocol. To use an encrypted protocol, use SSL instead of ldaps in the URL.

Property	Default Value	Description
AuthenticationMethod	Simple	The authentication method to use for all authentication requests into LDAP. Legal values are generally the same as those of the java.naming.security.authentication JNDI property. Choose one of: • simple — For clear-text password authentication. • DIGEST-MD5 — For more secure hashed password authentication. This method requires that the server use plain text password storage and only works with JRE 1.4 or later.
AuthenticationFilter	For most LDAP servers: (&(uid={uid})) (object-class=person)) or For Active Directory e-mail lookups: (&(userPrincipalName={uid})) (object-class=user)) [ActiveDirectory] For Active Directory Windows user name lookups: (&(sAMAccountName={uid}))(object-class=user))	The filter to use when looking up the user. When performing a user name based lookup, this filter is used to determine the LDAP entry that matches the supplied user name. The string "{uid}" in the filter is replaced with the supplied user name. Note: When you use this property to authenticate a user in SAP Control Center: • The property value should not contain any special characters, as listed above, in any of the common names or distinguished names. • Do not use Chinese or Japanese characters in user names or passwords of this property.

Property	Default Value	Description
AuthenticationScope	onelevel	The authentication search scope. The supported values for this are: • onellevel • subtree If you do not specify a value or if you specify an invalid value, the default value is used.
AuthenticationSearchBase	None	The search base used to authenticate users. If this property is not configured, the value for Default-SearchBase is used. Note: When you configure this property in the "admin" security configuration used to authenticate the administrator in SAP Control Center, the property value should not contain any special characters, as listed above, in any of the common names or distinguished names.
BindDN	None	The user DN to bind against when building the initial LDAP connection. In many cases, this user may need read permissions on all user records. If you do not set a value, anonymous binding is used. Anonymous binding works on most servers without additional configuration.

Property	Default Value	Description
BindPassword	None	The password for BindDN, which is used to authenticate any user. BindDN and BindPassword separate the LDAP connection into units. The AuthenticationMethod property determines the bind method used for this initial connection. If you configure the BindPassword from SAP Control Center, encrypted=true is set by default in the CSI configuration file. In addition, SAP recommends that you configure security providers from SAP Control Center, and not by manually editing the CSI configuration file. An encrypted BindPassword looks like this: <pre><options name="BindPassword" encrypted="true" value="1snjikf-wregfqr43hu5io..." /></pre> If you do not encrypt BindPassword, the option might look like this: <pre><options name="BindPassword" value="s3cr3T" /></pre>

Property	Default Value	Description
RoleSearchBase	None	The search base used to retrieve lists of roles. If this property is not configured, the value for DefaultSearchBase is used. Note: Setting the RoleSearchBase to the root in Active Directory (for example "DC=example,DC=com") results in a PartialResultsException error when validating the configuration or authenticating a user. To confirm, verify that example.com:389 is reachable. The DNS lookup may successfully resolve example.com to an IP address but port 389 may not be open with an Active Directory server listening on that port. In this case, adding an entry to the systemroot\system32\drivers\etc\hosts (typically, C:\WINDOWS\system32\drivers\etc\hosts) file on the machine where SAP Mobile Platform is installed resolves any communication error. Note: When you configure this property in the "admin" security configuration used to authenticate the administrator in SAP Control Center, the property value should not contain any special characters, as listed above, in any of the common names or distinguished names.

Property	Default Value	Description
RoleFilter	For SunONE/iPlanet: <code>(&(object-class=ldapsubentry) (object-class=nsroledefinition))</code> For Netscape Directory Server: <code>((object-class=groupofnames) (object-class=groupofunique-names))</code> For ActiveDirectory: <code>((object-class=groupofnames) (object-class=group))</code>	The role search filter. This filter should, when combined with the role search base and role scope, return a complete list of roles within the LDAP server. There are several default values, depending on the chosen server type. If the server type is not chosen and this property is not initialized, no roles are available. Note: When you use this property to authenticate a user in SAP Control Center: • The property value should not contain any special characters, as listed above, in any of the common names or distinguished names. • Do not use Chinese or Japanese characters in user names or passwords of this property.
RoleMemberAttributes	For Netscape Directory Server and OpenLDAP Server: member,unique-member	A comma-separated list of role attributes from which LDAP derives the DNs of users who have this role. These values are cross-referenced with the active user to determine the user's role list. One example of the use of this property is when using LDAP groups as placeholders for roles. This property has a default value only when the Netscape server type is chosen.
RoleNameAttribute	cn	The attribute of the role entry used as the role name in SAP Mobile Platform. This is the role name displayed in the role list or granted to the authenticated user.
RoleScope	onelevel	The role search scope. Supported values include: • onellevel • subtree If you do not specify a value or if you specify an invalid value, the default value is used.

Property	Default Value	Description
SkipRoleLookup	false	Set this property to true to grant the roles looked up using the attributes specified by the property UserRoleMembershipAttributes without cross-referencing them with the roles looked up using the RoleSearchBase and RoleFilter. LDAP configuration validation succeeds even when an error is encountered when listing all the available roles. The error is logged to the server log during validation but not reported in SAP Control Center, allowing the configuration to be saved. This has an impact when listing the physical roles for role mapping as well as in SAP Control Center. To successfully authenticate the user, set the SkipRoleLookup property to true.
UserRoleMembershipAttributes	For iPlanet/SunONE: nsRoleDN For Active Directory: memberOf For all others: none	Defines a user attribute that contains the DN's of all of the roles a user is a member of. These comma-delimited values are cross-referenced with the roles retrieved in the role search base and search filter to generate a list of user's roles. If SkipRoleSearch property is set to true, these comma-delimited values are not cross-referenced with the roles retrieved in the role search base and role search filter. See <i>Skipping LDAP Role Lookups (SkipRoleLookup)</i> in <i>Security</i>. Note: If you use nested groups with Active Directory, you must set this property to tokenGroups. See <i>LDAP Nested Groups and Roles in LDAP</i> in <i>Security</i>.
UserFreeformRoleMembershipAttributes	None	The freeform role membership attribute list. Users who have attributes in this comma-delimited list are automatically granted access to roles whose names are equal to the attribute value. For example, if the value of this property is department and user's LDAP record has the following values for the department attribute, { sales, consulting }, then the user will be granted roles whose names are sales and consulting.

Property	Default Value	Description
Referral	ignore	The behavior when a referral is encountered. Valid values are dictated by LdapContext, for example, follow, ignore, throw.
DigestMD5Authentication-Format	DN For OpenLDAP: User name	The DIGEST-MD5 bind authentication identity format.
UseUserAccountControlAttribute	For Active Directory: true	When this property is set to true, the UserAccountControl attribute is used to detect if a user account is disabled, if the account has expired, if the password associated with the account has expired, and so on. Active Directory uses this attribute to store this information.
controlFlag	optional	When you configure multiple authentication providers, use controlFlag for each provider to control how the authentication providers are used in the login sequence. controlFlag is a generic login module option rather than an LDAP configuration property. For more information, see <i>controlFlag Attribute Values</i> in <i>Security</i>.
EnableLDAPConnection-Trace	False	Enables LDAP connection tracing. The output is logged to a file in the temp directory. The location of the file is logged to the server log.
ConnectTimeout	0	Specifies the timeout, in milliseconds, when connecting to the LDAP server. The property value sets the JNDI com.sun.jndi.ldap.connect.timeout property, when attempting to establish a connection to a configured LDAP server. If the LDAP provider cannot establish a connection within the configured interval, it aborts the connection attempt. An integer less than or equal to zero results in the use of the network protocol's timeout value.

Property	Default Value	Description
ReadTimeout	0	Controls the length of time, in milliseconds, the client waits for the server to respond to a read attempt after the initial connection to the server has been established. The property values sets the JNDI <code>com.sun.jndi.ldap.read.timeout</code> property, when attempting to establish a connection to a configured LDAP server. If the LDAP provider does not receive an LDAP response within the configured interval, it aborts the read attempt. The read timeout applies to the LDAP response from the server after the initial connection is established with the server. An integer less than or equal to zero indicates no read timeout is specified.
LDAPPoolMaxActive	8	Caps the number of concurrent LDAP connections to the LDAP server. A non-positive value indicates no limit. If this option is set for multiple LDAP providers, the value set by the first LDAP provider loaded takes precedence over all the others. When LDAPPoolMaxActive is reached, any further attempts by the LDAP provider classes to borrow LDAP connections from the pool are blocked indefinitely until a new or idle object becomes available in the pool. Connection pooling improves the LDAP provider's performance and resource utilization by managing the number of TCP connections established with configured LDAP servers. A separate pool is associated with different SAP Mobile Platform security configurations, ensuring that the LDAP connections in the connection pool for a particular security configuration are isolated from any changes occurring outside this security configuration. A separate pool also ties the connection pool life cycle to that of the security configuration.

Property	Default Value	Description
providerDescription	None	(Optional). When enabled, allows the administrator to associate a description with the provider instance. Using a provider description makes it easier to differentiate between multiple instances of the same provider type: for example, when you have multiple login modules of the same type stacked in a security configuration, each targeting a different repository.

RADIUS Security Provider

The RADIUS security provider implements token-driven authentication against an existing RADIUS server.

The RADIUS provider fulfills authentication services only with the `com.sybase.security.radius.RadiusLoginModule` class; authorization and attribution services are not supported. If you need to configure these providers, you can use the LDAP providers with the RADIUS authentication provider.

RADIUS is not recommended for device user authentication: the application would require a different password each time it connects (generated by a token card), which would then require frequent user input to update that token password. Therefore RADIUS authentication should be limited to administration user authentication at most.

Note: The RADIUS authentication provider relies on third-party components that are not distributed by SAP. See the *SourceForge/JRadius* Web site for more information. If you need to configure the RADIUS provider, copy the third-party library into the `SMP_HOME\UnwiredServer\lib\3rdparty` folder, and restart SAP Control Center and SAP Mobile Server.

NTProxy Configuration Properties

(Not applicable to Online Data Proxy) Configure these properties to allow the operating system's security mechanisms to validate user credentials using NTProxy (Windows Native OS). Access these properties from the Authentication tab of the Security node in SAP Control Center.

Table 14. Authentication properties

Properties	Default Value	Description
Extract Domain From User Name	true	If set to true, the user name can contain the domain in the form of <username>@<domain>. If set to false, the default domain (described below) is always used, and the supplied user name is sent to through SSPI untouched.
Default Domain	The domain for the host computer of the Java Virtual Machine.	Specifies the default host name, if not overridden by the a specific user name domain.
Default Authentication Server	The authentication server for the host computer of the Java Virtual Machine.	The default authentication server from which group memberships are extracted. This can be automatically determined from the local machine environment, but this property to bypass the detection step.
Use First Password	false	If set to true, the login module attempts to retrieve only the user name and password from the shared context. It never calls the callback handler.
Try First Password	false	If set to true, the login module first attempts to retrieve the user name and password from the shared context before attempting the callback handler.
Clear Password	false	If set to true, the login module clears the user name and password in the shared context when calling either commit or abort.
Store Password	false	If set to true, the login module stores the user name and password in the shared context after successfully authenticating.
failAuthenticationIfNoRoles	false	If set to true, the provider fails to authenticate the user if the user is not assigned any physical roles.

Properties	Default Value	Description
providerDescription	none	(Optional). When enabled, allows the administrator to associate a description with the provider instance. Using a provider description makes it easier to differentiate between multiple instances of the same provider type: for example, when you have multiple login modules of the same type stacked in a security configuration, each targeting a different repository.

NoSecurity Configuration Properties

A NoSecurity provider offers pass-through security for SAP Mobile Server, and should be typically be reserved for development or testing. SAP strongly encourages you to avoid using this provider in production environments — either for administration or device user authentication.

- The NoSecLoginModule class provides open authentication services
- The NoSecAuthorizer class provides authorization services
- The NoSecAttributer provides attribution services

You need to configure only the authentication properties for the NoSecurity provider.

Table 15. Authentication Properties

Property	Default Value	Description
useUsernameAsIdentity	true	If this option is set to true, the user name supplied in the callback is set as the name of the principal added to the subject.
identity	nosec_identity	The value of this configuration option is used as the identity of the user if either of these conditions is met: • No credentials were supplied. • The useUsernameAsIdentity option is set to false.

Property	Default Value	Description
Use First Password	false	If set to true, the login module attempts to retrieve only the user name and password from the shared context. It never calls the callback handler.
Try First Password	false	If set to true, the login module first attempts to retrieve the user name and password from the shared context before attempting the callback handler.
Clear Password	false	If set to true, the login module clears the user name and password in the shared context when calling either commit or abort.
Store Password	false	If set to true, the login module stores the user name and password in the shared context after successfully authenticating.
providerDescription	none	(Optional). When enabled, allows the administrator to associate a description with the provider instance. Using a provider description makes it easier to differentiate between multiple instances of the same provider type: for example, when you have multiple login modules of the same type stacked in a security configuration, each targeting a different repository.

Note: When you create a new security configuration, SAP Mobile Platform sets the NoSecurity provider by default. SAP recommends that after you add, configure, and validate your providers, you remove the NoSecurity provider. For more information, see *Creating a Security Configuration*.

Certificate Authentication Properties

This authentication provider is deprecated. Use CertificateValidationLoginModule instead. Add and configure authentication provider properties for CertificateAuthenticationLoginModule, or accept the default settings.

Note: This provider cannot be used for administrative security (in the "admin" security configuration).

Table 16. CertificateAuthenticationLoginModule properties

Property	Description
Implementation class	The fully qualified class that implements the login module. <code>com.sybase.security.core.CertificateAuthenticationLoginModule</code> is the default class.
Provider type	<code>LoginModule</code> is the only supported value.
Control flag	Determines how success or failure of this module affects the overall authentication decision. <code>optional</code> is the default value.
Clear password	(Optional) If true, the login module clears the user name and password from the shared context. The default is false.
Store password	(Optional) If true, the login module stores the user name and password in the shared context. The default is false.
Try first password	(Optional) If true, the login module attempts to retrieve user name and password information from the shared context, before using the callback handler. The default is false.
Use first password	(Optional) If true, the login module attempts to retrieve the user name and password only from the shared context. The default is false.
Enable revocation checking	(Optional) Enables online certificate status protocol (OCSP) certificate checking for user authentication. If you enable this option, you must enable OCSP in SAP Mobile Server. This provider uses the values that are defined in the SSL security profile. Revoked certificates result in authentication failure when: • Revocation checking is enabled, and • OCSP properties are configured correctly.
Regex for username certificate match	(Optional) By default, this value matches that of the certificates common name (CN) property used to identify the user. If a mobile application user supplies a user name that does not match this value, authentication fails.

Property	Description
Trusted certificate store	(Optional) The file containing the trusted CA certificates (import the issuer certificate into this certificate store). Use this property and <code>Store Password</code> property to keep the module out of the system trust store. The default SAP Mobile Server system trust store is <code>SMP_HOME\Servers\UnwiredServer\Repository\Security\keystore.jks</code>. If you do not specify a store location: • SAP Mobile Server checks to see if a store used by the JVM (that is, the one defined by the <code>javax.net.ssl.trustStoreType</code> system property). • If the system property is not defined, then this value is used: <code>{java.home}/lib/security/jssecacerts</code> • If that location also doesn't exist, then this value is used: <code>{java.home}/lib/security/cacerts</code> Note: This property is required only if <code>Validate certificate path</code> is set to true.
Trusted certificate store password	(Optional) The password required to access the trusted certificate store. For example, import the issuer of the certificate you are trying to authenticate into the shared JDK cacerts file and specify the password using this property. Note: This property is required only if <code>Validate certificate path</code> is set to true. However, you do not need to configure this value if the default is used. The default value is the value of the <code>javax.net.ssl.trustStorePassword</code> property.
Trusted certificate store provider	(Optional) The keystore provider. For example, "SunJCE." Note: This property is required only if <code>Validate certificate path</code> is set to true. However, you do not need to configure this value if the default is used. The default value is the value of the <code>javax.net.ssl.trustStoreProvider</code> property. If it is not defined, then the most preferred provider from the list of registered providers that supports the specified certificate store type is used.

Property	Description
Trusted certificate store type	(Optional) The type of certificate store. For example, "JKS." Note: This property is required only if Validate certificate path is set to true. However, you do not need to configure this value if the default is used. The default value is the value of the <code>javax.net.ssl.trustStore</code> property. If this value is not defined, then default value is the keystore type as specified in the Java security properties file, or the string "jks" (Java keystore) if no such property exists.
Validate certificate path	If true (the default), performs certificate chain validation of the certificate being authenticated, starting with the certificate being validated. Verifies that the issuer of that certificate is valid and is issued by a trusted certificate authority (CA), if not, it looks up the issuer of that certificate in turn and verifies it is valid and is issued by a trusted CA. In other words, it builds up the path to a CA that is in the trusted certificate store. If the trusted store does not contain any of the issuers in the certificate chain, then path validation fails.
providerDescription	(Optional). When enabled, allows the administrator to associate a description with the provider instance. Using a provider description makes it easier to differentiate between multiple instances of the same provider type: for example, when you have multiple login modules of the same type stacked in a security configuration, each targeting a different repository.

Certificate Validation Properties

Add and configure provider properties for `CertificateValidationLoginModule`, or accept the default settings. `CertificateValidationLoginModule` can be used in conjunction with other login modules that support certificate authentication (for example, `LDAPLoginModule`) by configuring `CertificateValidationLoginModule` before the login modules that support certificate authentication.

You can only use this provider to validate client certificates when an HTTPS listeners is configured to use mutual authentication.

Table 17. CertificateValidationLoginModule properties

Property	Description
Implementation class	The fully qualified class that implements the login module. <code>com.sybase.security.core.CertificateValidationLoginModule</code> is the default class.
crl.[index].uri	Specifies the universal resource identifier for the certificate revocation list (CRL). Multiple CRLs can be configured using different values for the index. The CRLs are processed in index order. For example: <pre>crl.1.uri=http://crl.verisign.com/ThawtePersonalFreemailIssuingCA.crl crl.2.uri=http://crl-server/</pre>
Provider type	LoginModule is the only supported value.
Validated certificate is identity	(Optional) Determines if the certificate should be set the authenticated subject as the user ID. If the CertificateValidationLoginModule is used in conjunction with other login modules that establish user identity based on the validated certificate, set this value to <code>false</code> . If you are implementing this provider with a DCN security configuration, and it's also not used with SSO, then set this property to <code>true</code> . <code>False</code> is the default value.
Enable revocation checking	(Optional) Enables online certificate status protocol (OCSP) certificate checking for user authentication. If you enable this option, you must enable OCSP in SAP Mobile Server. This provider uses the values that are defined in the SSL security profile. Revoked certificates result in authentication failure when: • Revocation checking is enabled, and • OCSP properties are configured correctly.

Property	Description
Trusted certificate store	(Optional) The file containing the trusted CA certificates (import the issuer certificate into this certificate store). Use this property and <code>Store Password</code> property to keep the module out of the system trust store. The default SAP Mobile Server system trust store is <code>SMP_HOME\Servers\UnwiredServer\Repository\Security\keystore.jks</code>. If you do not specify a store location: - SAP Mobile Server checks to see if a store used by the JVM (that is, the one defined by the <code>javax.net.ssl.trustStoreType</code> system property). - If the system property is not defined, then this value is used: <code>{java.home}/lib/security/jssecacerts</code> - If that location also doesn't exist, then this value is used: <code>{java.home}/lib/security/cacerts</code> Note: This property is required only if <code>Validate certificate path</code> is set to true.
Trusted certificate store password	(Optional) The password required to access the trusted certificate store. For example, import the issuer of the certificate you are trying to authenticate into the shared JDK cacerts file and specify the password using this property. Note: This property is required only if <code>Validate certificate path</code> is set to true. However, you do not need to configure this value if the default is used. The default value is the value of the <code>javax.net.ssl.trustStorePassword</code> property.
Trusted certificate store provider	(Optional) The keystore provider. For example, "SunJCE." Note: This property is required only if <code>Validate certificate path</code> is set to true. However, you do not need to configure this value if the default is used. The default value is the value of the <code>javax.net.ssl.trustStoreProvider</code> property. If it is not defined, then the most preferred provider from the list of registered providers that supports the specified certificate store type is used.

Property	Description
Trusted certificate store type	(Optional) The type of certificate store. For example, "JKS." Note: This property is required only if Validate certificate path is set to true. However, you do not need to configure this value if the default is used. The default value is the value of the <code>javax.net.ssl.trustStore</code> property. If this value is not defined, then default value is the keystore type as specified in the Java security properties file, or the string "jks" (Java keystore) if no such property exists.
Validate certificate path	If true (the default), performs certificate chain validation of the certificate being authenticated, starting with the certificate being validated. Verifies that the issuer of that certificate is valid and is issued by a trusted certificate authority (CA), if not, it looks up the issuer of that certificate in turn and verifies it is valid and is issued by a trusted CA. In other words, it builds up the path to a CA that is in the trusted certificate store. If the trusted store does not contain any of the issuers in the certificate chain, then path validation fails.
providerDescription	(Optional). When enabled, allows the administrator to associate a description with the provider instance. Using a provider description makes it easier to differentiate between multiple instances of the same provider type: for example, when you have multiple login modules of the same type stacked in a security configuration, each targeting a different repository.

HTTP Basic Authentication Properties

The `HttpAuthenticationLoginModule` provider authenticates the user with given credentials (user name and password) against the secured Web server using a GET against a URL that requires basic authentication, and can be configured to retrieve a cookie with the configured name and add it to the JAAS subject to facilitate single sign-on (SSO).

Configure this provider to authenticate the user by:

- Using only the specified user name and password.
- Using only the specified client value or values.
- Attempting token authentication. If that fails, revert to basic authentication using the supplied user name and password. You may find this helpful when using the same security configuration for authenticating users with a token, such as device users hitting the network edge, and when DCN requests from within a firewall present a user name and password but no token.

Note: The `HttpAuthenticationLoginModule` allows token validation by connecting to an HTTP server capable of validating the token specified in the HTTP header and cookie set in the session.

Table 18. HttpAuthenticationLoginModule Configuration Options

Configuration Option	Default Value	Description
URL	None	The HTTP or HTTPS URL that authenticates the user. For SSO, this is the server URL from which SAP Mobile Server acquires the SSO cookie/token.
Disable server certificate validation	False	(Optional) The default is false. If set to true, this property disables server certificate validation when establishing an HTTPS connection to the SWS using the configured URL. Set to true only for configuration debugging.
SSO cookie name	None	(Optional) Name of the cookie set in the session between the LoginModule and the secured Web server, and holds the SSO token for single sign-on. The provider looks for this cookie in the connection to the secured Web server. If the cookie is found, it is added to the authenticated subject as a named credential. The authentication provider ignores the status code when an SSO cookie is found in the session; authentication succeeds regardless of the return status code.
Roles HTTP header	None	(Optional) The name of an HTTP header that the server may return. The header value contains a comma-separated list of roles to be granted.

Configuration Option	Default Value	Description
Successful connection status code	200	HTTP status code interpreted as successful when connection is established to the secured Web server.
HTTP connection timeout interval	60000 milliseconds (60 seconds)	The value, in milliseconds, after which an HTTP connection request to the Web-based authentication service times out. If the HTTP connection made in this module (for either user authentication or configuration validation) does not have a timeout set, and attempts to connect to a Web-based authentication service that is unresponsive, the connection also becomes unresponsive, which could potentially cause SAP Mobile Server to become unresponsive. Setting the timeout interval ensures that authentication failure is reported without waiting indefinitely for the server to respond.
SendClientHttpValuesAs	None	Comma-separated list of strings that indicate how to send ClientHttpValuesToSend to the HTTP server. For example: <pre>SendClientHttpValuesAs=header:header_name, cookie: cookie_name</pre> This property does not apply if the user is to be authenticated using only the supplied user name and password .

Configuration Option	Default Value	Description
ClientHttpValuesToSend	None	A comma-separated list of client HTTP values to be sent to the HTTP server. For example: <pre>ClientHttpValuesToSend=client_personalization_key,client_cookie_name</pre> Set this property if you are using token authentication. Setting this property triggers token authentication. Only token authentication is attempted, unless TryBasicAuthIfTokenAuthFails is configured to true in conjunction with ClientHttpValuesToSend. This property does not apply if the user is to be authenticated using only the supplied user name and password .

Configuration Option	Default Value	Description
SendPasswordAsCookie	None	Deprecated. Use only for backward compatibility. New configurations should configure token authentication using SendClientHttpValuesAs and ClientHttpValuesToSend. Sends the password to the URL as a cookie with this name. If not specified, the password is not sent in a cookie. This property is normally used when there is a cookie-based SSO mechanism in use (for example, SiteMinder), and the client has put an SSO token into the password. The token can be propagated from the personalization keys and HTTP header and cookies to the secured Web server without impacting the password field.
TryBasicAuthIfTokenAuthFails	False	Specifies whether the provider should attempt basic authentication using the specified user name and password credentials if token authentication is configured and fails. This property is applicable only if token authentication is enabled. This property does not apply if the user is to be authenticated using only the supplied user name and password .

Configuration Option	Default Value	Description
UsernameHttpHeader	None	HTTP response header name returned by the HTTP server with the user name retrieved from the token. Upon successful authentication, the retrieved user name is added as a SecNamePrincipal. This property does not apply if the user is to be authenticated using only the supplied user name and password .
regexForUsernameMatch	None	Regular expression used for matching the supplied user name with the user name returned by the HTTP server in the UsernameHttpHeader. The string "{username}" in the regex is replaced with the specified user name before using it. If specified, it matches the user name retrieved from the UsernameHttpHeader to the user name specified in the callback handler. If the user names do not match, authentication fails. If the user names match, both the specified user name and the retrieved user name are added as SecNamePrincipals to the authenticated subject. This property does not apply if the user is to be authenticated using only the supplied user name and password .

Configuration Option	Default Value	Description
TokenExpirationTimeHttpHeader	None	HTTP response header name that is returned by the HTTP server with the validity period of the token in milliseconds since the start of January 1, 1970. If the header is returned in the HTTP response from the secured Web server, the token is cached for the duration it remains valid unless TokenExpirationInterval is also configured. If this response header is not returned with the token, it might result in unintended use of the token attached to the authenticated context even after it has expired. This property does not apply if the user is to be authenticated using only the supplied user name and password .

Configuration Option	Default Value	Description
TokenExpirationInterval	0	Specifies the interval, in milliseconds to be deducted from the actual expiration time returned in TokenExpirationTimeHttpHeader. This ensures that the token credential retrieved from the authenticated session remains valid until it is passed to the SWS for single sign-on to access MBOs. Note: This property does not apply if the user should be authenticated using only the supplied user name and password. Note: If the configured TokenExpirationInterval value exceeds the amount of time the token is valid, authentication by the HttpAuthenticationLoginModule fails even if the token is validated successfully by the secured Web server.
CredentialName	None	Name to set in the authentication credential that contains the token returned in SSOCookieName. If this property is not configured, the SSOCookieName is set as the name of the token credential.

Configuration Option	Default Value	Description
providerDescription	None	(Optional). When enabled, allows the administrator to associate a description with the provider instance. Using a provider description makes it easier to differentiate between multiple instances of the same provider type: for example, when you have multiple login modules of the same type stacked in a security configuration, each targeting a different repository.

SAP SSO Token Authentication Properties

The SAPSSOTokenLoginModule has been deprecated. Use the HttpAuthenticationLoginModule when SAP SSO2 token authentication is required.

Preconfigured User Authentication Properties

The PreConfiguredUserLoginModule authenticates the SAP Mobile Platform Administrator user whose credentials are specified during installations.

This login module is recommended only to give the Platform administrator access to SAP Control Center so it can be configured for production use. Administrators are expected to replace this login module immediately upon logging in for the first time. For details on how to setup administrator authentication in a production deployment, see *Enabling Authentication and RBAC for Administrator Logins* in the *Security* guide.

The PreConfiguredUserLoginModule:

- Provides role based authorization by configuring the provider `com.sybase.security.core.RoleCheckAuthorizer` in conjunction with this authentication provider.
- Authenticates the user by comparing the specified user name and password against the configured user. Upon successful authentication, the configured roles are added as Principals to the Subject.

Table 19. PreConfiguredUserLoginModule properties

Property	Description
User name	A valid user name. Do not use any of these restricted special characters: , = : ' " * ? &.

Property	Description
Password	The encoded password hash value.
Roles	Comma separated list of roles granted to the authenticated user for role-based authorization. Platform roles include SUP Administrator, SUP Domain Administrator, and SUP Helpdesk. Roles are mandatory for "admin" security configuration. For example, if you define SUP Administrator to this property, the login ID in the created login module has Platform administrator privileges. The SUP Helpdesk role has the fewest privileges. If multiple roles are defined for this property, a role with more privileges (SUP Administrator or SUP Domain Administrator) is used for authorizing users. Note: If you use other values, ensure you map SAP Mobile Platform roles to the one you define here.
providerDescription	(Optional). When enabled, allows the administrator to associate a description with the provider instance. Using a provider description makes it easier to differentiate between multiple instances of the same provider type: for example, when you have multiple login modules of the same type stacked in a security configuration, each targeting a different repository.

Audit Provider Properties

The security configuration for SAP Mobile Platform includes an audit provider with three components: audit filter, audit formatter, and audit destination.

An auditor consists of one destination, one filter, and one formatter.

- The supported value for destination is `com.sybase.security.core.FileAuditDestination`. Optionally, you can develop a custom provider and configure it as the audit destination, formatter, and filter. See *CSI Audit Generation and Configuration* in *Security*.
- The only supported value for the filter is `com.sybase.security.core.DefaultAuditFilter`.
- The only supported value for the formatter is `com.sybase.security.core.XmlAuditFormatter`.

For information on developing a custom provider and configuring it as the audit destination, formatter, and filter, see *Security API* in *Developer Guide: SAP Mobile Server Runtime*.

For detailed information on the audit packages, see *Security Configuration* in *Developer Guide: SAP Mobile Server Runtime > Management API> Client Metadata*.

DefaultAuditFilter Properties

The audit filter component configures the resource classes for which the audit records should be routed to the associated destination.

Filter resource classes require a specific syntax. The audit token identifies the source for core audit requests of operations, such as auditing the results for authorization and authentication decisions, in addition to placing information such as active provider information into the audit trail. The audit records have their resource class prefixed by the prefix core. The CSI core is able to audit multiple items.

DefaultAuditFilter Configuration Properties

The property name default value description is:

```
(1)caseSensitiveFiltering false set to true to use case sensitivity
when matching resource classes and actions
(2)filter
default
value="(ResourceClass=core.subject,Action=authorization.role)
(ResourceClass=core.subject,Action=authorization.resource)
(ResourceClass=core.subject,Action=authentication)
(ResourceClass=core.subject,Action=logout)
(ResourceClass=core.profile)
(ResourceClass=providers.*) (ResourceClass=clients.*) "
description = the filter string that determines whether an audit
record should be written out to the audit destination.
```

Syntax

Filter resource classes consist of one or more filter expressions that are delimited by parenthesis (). Square brackets ([]) denote optional values. The syntax is:

```
[key1=value [,key2=value...]].
```

The allowed keys are: ResourceClass, Action, or Decision.

This table describes core auditable items:

Resource Class	Action	Description	Attributes
provider	activate	Called when a provider is activated by CSI. The resource ID is the provider class name.	Generated unique provider identifier.

Resource Class	Action	Description	Attributes
subject	authentication.provider	The result of a provider's specific authentication request. Depending on the other providers active, the actual CSI request for authentication may not reflect this same decision. This resource class is not a provider-generated audit record. CSI core generates this audit record automatically after receiving the provider's decision. The resource ID is not used.	• Provider identifier • Decision (yes or no) • Failure reason (if any) • Context ID
subject	authentication	The aggregate decision after considering each of the appropriate provider's authentication decisions. This record shares the same request identifier as the corresponding authentication provider records. The resource ID is the subject identifier if authentication is successful.	• Decision (yes or no) • Context ID
subject	authorization.role.provider	The result of a provider's specific role authorization request. The resource ID is the subject ID.	• Provider identifier • Decision (yes, no or abstain) • Role name • Supplied subject identifier, if different from context subject • Context ID

Resource Class	Action	Description	Attributes
subject	authorization.role	The result of a resource-based authorization request. The resource ID is the subject ID.	- Resource name - Access requested - Decision (yes or no) - Supplied subject identifier, if different from context subject - Context ID
subject	logout	Generated when an authenticated context is destroyed. The resource ID is the subject ID.	Context ID
subject	create.provider	Provider-level record issued for anonymous self-registration requests. The resource ID is the subject identifier.	- Provider identifier - Decision - Subject attributes
subject	create	Aggregate, generated when an anonymous self-registration request is made. The resource ID is the subject identifier.	- Decision - Subject attributes
subject	authorization.resource	The aggregate authorization decision, which is made after considering each of the appropriate provider's result. The resource ID is the subject ID.	- Resource ID - Access requested - Decision (yes or no) - Subject ID supplied, if different from context subject - Context ID

Examples

- **Example 1** – enables auditing of all the CSI core resource classes that involve a deny decision:

```
(ResourceClass=core.*,Decision=Deny)
```

- **Example 2** – enables auditing for all core resource classes where the action is the subject modification action:

```
Resource=core.*,Action=subject.modify.*)
```

FileAuditDestination Properties

The FileAuditDestination is a simple file-based provider that logs the audit records to a file which is rolled over upon reaching a specified size.

This provider can safely share access to a file between multiple instances as long as they are all in the same VM. To integrate with a customer's existing audit infrastructure, a custom audit destination provider can be developed and deployed. See *com.sybase.security.core.FileAuditDestination* in *Developer Guide: SAP Mobile Server Runtime*.

Table 20. File Audit Destination Configuration Options

Configuration Option	Description
auditFile	The absolute path of the file to write the audit records.
encoding	The character encoding used when writing the audit data (default=UTF-8).
logSize	This option may be supplied to specify the maximum audit log file size before a rollover occurs.
compressionThreshold	This option may be supplied to specify the number of uncompressed audit log rollover files that are created, before compression is used to archive the audit data.
deleteThreshold	This option may be supplied to specify the number of audit log files that will be preserved. This value includes the main audit log, so a value of "3" allows an audit.log, audit.log.0 and audit.log.1 before deleting old logs.

Configuration Option	Description
errorThreshold	This option may be supplied to specify the maximum number of audit log files that are allowed. When this threshold is reached, an error occurs and all auditing fails. For example, with this value set to "3", audit.log, audit.log.0 and audit.log.1 will be created according to the maximum log size value. If another audit log rollover is triggered, then all audit operations will fail until one of the rollover files is removed. This value is mutually exclusive with the deletion threshold, and the smallest value of the two takes effect.

XMLAuditFormatter Properties

The audit formatter component formats an audit record from its component parts. An audit formatter is supplied to the active audit destination upon initialization, where the audit destination can use the formatter if required.

The default provider `com.sybase.security.core.XmlAuditFormatter` formats audit data into an XML record. The audit records generated by this provider are of the format

```
<AuditRecord Action="[action]" Decision="[decision]"
When="[timestamp]"> <Resource Class="[resource classname]"
ID="[resource id]" /> <Attribute Name="[attribute1 name]"
Value="[attribute1 value]" /> <Attribute Name="[Map attribute name]"
Key="[Map Key1 name]" Value="[Map value associated with the key1]" />
<Attribute Name="[Map attribute name]" Key="[Map Key2 name]"
Value="[Map value associated with the key2]" /> <Attribute
Name="[List attribute name]" Value="[List value1]" /> <Attribute
Name="[List attribute name]" Value="[List value2]" /> </AuditRecord>
```

Creating Logical Roles for a Security Configuration

Use logical roles to control access and/or group a large number of users who use the same security configuration.

Create the required logical roles and map them to one or more physical role, then assign both a security configuration and a logical role to an application (through the application connection template). Users must have one of the physical roles that the logical role is mapped to in order to access the application.

1. In the left navigation pane of SAP Control Center, select **Security** > **Security configuration name**.
2. In the right administration pane, click the **Role Mappings** tab.
3. Click **New**.
4. Enter a name for the logical role you wish to create.
5. Select one of the following options:

State	Description
AUTO	To map the logical role to a physical role of the same name.
NONE	To disable the logical role, which means that the logical role is not authorized.
MAP	To manually map the logical role when the physical and logical role names do not match. See <i>Mapping a Physical Role Manually</i> .

6. Click **OK**.

See also

- *Mapping a Physical Role Manually* on page 243
- *Security Configurations* on page 199

Mapping Roles for a Security Configuration

Map logical roles to physical roles for a security configuration by setting the mapping state. Role mappings performed at the security configuration level are applied to all domains that are assigned the security configuration.

Prerequisites

SAP Mobile Platform cannot query all enterprise security servers directly; to perform authentication successfully, know the physical roles that are required.

Task

1. In the left navigation pane of SAP Control Center, select **Security > Security configuration name**.
2. In the right administration pane, click the **Role Mappings** tab.
3. Select a logical role and select one of the following in the adjacent list:

State	Description
AUTO	To map the logical role to a physical role of the same name.
NONE	To disable the logical role, which means that the logical role is not authorized.
MAP	To manually map the logical role when the physical and logical role names do not match. See <i>Mapping a Physical Role Manually</i> .

Mapping a Physical Role Manually

Use the Role Mappings dialog to manually map required physical roles for a logical role when physical and logical role names do not match. If names do not match, the AUTO mapping state does not work.

Prerequisites

SAP Mobile Platform cannot query all supported enterprise security servers directly; for successful authentication, you must know the physical roles your back-end systems require.

Task

You can map a logical role to one or more physical roles. You can also map multiple logical roles to the same physical role. If a role does not exist, you can also add or delete names as needed.

1. Review the list of existing physical role names that you can map to the logical role you have selected. If the list retrieved is too long to locate the name quickly, either:
 - Click the banner of Available Roles list to sort names alphanumerically.
 - Start typing characters in the box, then click the Search button to filter the available list.
2. If a role that you require still does not appear, enter the **Role name** and click the + button. The role name appears in the **Available roles** list with an asterisk (*). This asterisk indicates that an available role was added by an administrator, not a developer.
3. To remove a role you no longer require from the **Available roles** list, select the name and click the x button adjacent to the **Role name** field.
The role is removed and can no longer be mapped to a logical role.
4. To map a logical role that appears in the text area of the Role Mappings dialog to a physical role:
 - a) Select one or more **Available roles**.
 - b) Click **Add**.
5. To unmap a role:
 - a) Select one or more **Mapped roles**.
 - b) Click **Remove**.
The roles are returned to the **Available roles** list.
6. Click **OK** to save these changes.

Once a logical role has been manually mapped, the mapping state changes to MAPPED. The roles you have mapped appear in the active Physical Roles cell for either a package-specific or server-wide role mappings table.

Mapping State Reference

The mapping state determines the authorization behavior for a logical name instance.

State	Description
AUTO	Map the logical role to a physical role of the same name. The logical role and the physical role must match, otherwise, authorization fails.
NONE	Disable the logical role, which means that the logical role is not authorized. This mapping state prohibits anyone from accessing the resource (MBO or Operation). Carefully consider potential consequences before using this option.
MAPPED	A state that is applied after you have actively mapped the logical role to one or more physical roles. Click the cell adjacent to the logical role name and scroll to the bottom of the list to see the list of mapped physical roles.

Mapping Logical Roles to a Technical User in a Repository

Mapping the SUP Push User or SUP DCN User logical role binds it to another physical user identity rather than a role.

1. In the navigation pane of SAP Control Center, select the security configuration that is assigned to the Push or DCN domain.
2. Locate the logical role.
3. Select **Map Role**.
4. In the Role Mapping dialog, enter the technical role name in the **Role name** box.
user:supuser, where *supuser* is the technical user in the repository.
5. Click **Add (+)**.
6. Select the newly added technical role.
Technical role can be based on basic authorization or certificate (client).
7. Click **Add**.

If you are using ActiveDirectory, and are using an e-mail address for the technical user names, the definition appears as
username@myaddress@DomainSecurityConfigName.

The logical role now shows the mapping state changes to MAPPED. In the **Physical Roles** drop-down, verify that the new technical role is added.

SUP Roles to Support EIS Operations: SUP DCN User and SUP Push User

SUP DCN User and SUP Push User roles are the mechanisms by which illicit EIS DCN or push notification operations are prevented. Like other built-in platform roles, SUP DCN User and SUP Push User are logical roles that are available to all new security configurations.

Before any DCN event is submitted, the person or group mapped to this role must be authorized (after first being authenticated) by a security provider defined as part of a named security configuration. Submitted DCN events that require authorization include:

- Cache updates
- Operation performance

The SUP Push user role is mandatory; with this role the EIS cannot deliver push notifications to SAP Mobile Server for a registered application connection. Before any push event is submitted by the EIS, the authenticated user performing the push must be authorized by being in the SUP Push User logical role. Push events that require authorization include:

- Triggering a Hybrid App package

You can choose different physical role mapping targets to authorize, or authenticate and authorize EIS events using the logical roles. Depending on the authorization method used, the implementation varies:

- **Certificate authorization** – SAP recommends that you use CertificateValidationLoginModule for maximum security. CertificateValidationLoginModule validates the user certificate passed during mutual certificate authentication. Unlike other methods, it confers no physical roles; therefore, the platform administrator must create a logical role mapping. Typically, the user has a certificate that includes a Subject distinguished name containing a common name (cn=TechnicalUser), so it creates a logical role mapping between the logical role and user:TechnicalUser in the CN. To implement certificate authorization, see *Setting Up Authorization with Certificate Validation* in *Security*.

Note: While explicitly mapping a certificate user name for SUP Push User role in SAP Control Center, ensure there is a space after every comma. Example: user : CN:PushTest, OU=SSL Server, O=SAP-AG, C=DE". If you are using push notification with strong mutual authentication, you can only use the Admin security configuration. Ensure you add a CertificateValidationLoginModule to the Admin security configuration and use it as the default security configuration in the push-enabled domain. If any other security configuration is used, a user not in Required role error is generated in the client log.

- **Technical user authorization** – If the role cannot be mapped to a real user in the security repository of the configured security provider used by the security configuration, you may need to create a new technical user or use an existing technical user for EIS operation role mappings. In this case, no authentication is required as the user is not a real user in the traditional sense. To implement technical user authorization, SAP recommends that you

create a security configuration that includes an LDAP provider. To implement technical user authentication, see *Setting Up Authorization with a Technical User Role Stored in a Repository* in *Security*.

- **Real user authorization** – (Applies only to DCN) if the role must be mapped to a real user, you can authenticate and authorize the user mapped to the SUP DCN User role. You can also use PreconfiguredUserLogin module to perform HTTP Basic authentication, where the module extracts the user information from the request parameter in a URL. To implement real user authentication, see *Setting Up Authorization with PreConfiguredUserLogin Values* in *Security*.

Once you have multiple providers configured, especially when implementing authorization with single sign-on, you can stack them so they are processed in correct order.

Assigning a Security Configuration to a Domain

Assign security configurations to one or more domains. This allows the supAdmin to offer a security repository for application user authentication and authorization, as well as to share security providers across domains in case one tenant uses multiple domains.

Prerequisites

A supAdmin must already have created one or more security configurations in the Security node of SAP Control Center.

Task

1. In the left navigation pane, navigate to **ClusterName > Domains > DomainName > Security**
2. In the right administration pane, select the **General** tab and click **Assign**.
3. From the list of available security configurations, select the appropriate configuration for domain security, and click **OK**.
If successful, an Assigned successfully message appears, and the newly added security configuration is listed in the domain-level Security node.

See also

- *Creating a Security Configuration* on page 200

Viewing Security Configuration Usage

You can view the security configurations and logical roles used to access applications. This enables you to assess the impact of changing a security configuration or logical role.

1. In the left navigation pane of SAP Control Center select **Security > Security configuration name**.
2. In the right administration pane, click the **Role Mappings** tab.

- To view the applications that the security configuration provides access to, click **Usage** and review the information on the **Domains and Packages** and **Application Connection Templates** tabs.
- To view the applications that a specific role in a security configuration provides access to, select a roles, click **Usage**, and review the information on the **Application Connection Templates** tab.

Note: The **Domains and Packages** tab lists all MBO packages that the security configuration provides access to, no matter what logical role is selected.

Anonymous Access Security Configuration

Allow unauthenticated users access to application data, for example, applications that allow users to browse a read-only product catalog without logging in by assigning the anonymous security configuration to the application.

The anonymous security configuration:

- Is a preconfigured user name/password login module that accepts the user name "anonymous" and the password "anonymous". User name is not case sensitive, but password is.
- Cannot be modified.
- Can be enabled when manually creating an application with anonymous access or when setting anonymous access for an existing application by creating an application connection template for it that uses the anonymous security configuration.

Agency Security Configuration

The Agency security configuration allows Agency application users to be authenticated by the Agency Server.

The Agency security configuration:

- Is a special-purpose security configuration assigned to the "default" domain.
- Uses the `NoSecLoginModule` and forwards login requests to the Agency server for authentication.
- Is automatically selected when creating Agency application definitions and cannot be changed.
- Is reserved for system use and should not be used directly.

SiteMinder Authentication with SAP Mobile Platform

Configure your SiteMinder environment for authentication in SAP Mobile Platform.

CA SiteMinder enables policy-based authentication and single sign-on with SAP Mobile Platform. You can configure SiteMinder and SAP Mobile Platform integration in a number of ways, depending on your environment. For detailed examples focusing on SiteMinder-specific configurations for SAP Mobile Platform, see *How-To: Set up SUP with SiteMinder* at <http://scn.sap.com/docs/DOC-29574>.

SiteMinder Client Authentication

SiteMinder provides various client authentication options for SAP Mobile Platform, including single sign-on (SSO), tokens, and Network Edge.

SiteMinder client authentication includes:

- Network Edge – when a reverse proxy or Relay Server in the DMZ is protected by SiteMinder, the SAP Mobile Platform client is challenged for basic authentication credentials. If the credentials are valid, an SMSESSION cookie is issued and the client is allowed through to the SAP Mobile Platform server. The client begins a session (RBS, MBS, or OData) by sending an HTTP(S) request to the reverse proxy. The reverse proxy detects the unauthenticated request, and challenges using basic authentication. After the 401 challenge, the client may already have network credentials configured, or executes a callback to prompt for credentials.
- Non-Network Edge – the Network Edge (reverse proxy or Relay Server) is not protected. The client's request is allowed to flow to SAP Mobile Platform, where a LoginModule presents the basic credentials to a SiteMinder-protected Web server on behalf of the client. SAP Mobile Platform server retains the SMSESSION cookie and credentials for the client.
- External tokens – the SAP Mobile Platform client application obtains an SMSESSION cookie external to the SAP Mobile Platform libraries using custom application processing. This SMSESSION token passes into the SAP Mobile Platform libraries as a cookie. SAP Mobile Platform libraries add the cookie to subsequent HTTP requests to SAP Mobile Platform server. The cookie may or may not be checked at the Network Edge.
- SAP SSO2 integration – the SAP Mobile Platform user is initially authenticated by SiteMinder, resulting in an SMSESSION for the user. This SMSESSION is forwarded along with the SAP user ID to a SiteMinder SAP agent running inside of NetWeaver as a LoginModule. The SMSESSION is revalidated, and the TokenIssuingLoginModule is allowed to issue an SSO2 ticket for the specified SAP user ID. This ticket returns to SAP Mobile Platform as a MYSAPSSO2 cookie. SAP Mobile Platform now has both an SMSESSION and an SSO2 ticket to use for SSO purposes with various EIS depending on which SSO mechanism the EIS requires.

Note: In any of these authentication patterns, you can add the SMSESSION token as a credential to the authenticated SAP Mobile Platform subject for use in single sign-on to SiteMinder-protected systems.

Single Sign-on to a SiteMinder-protected EIS

SiteMinder single sign-on (SSO) provides integration between a SiteMinder-protected EIS and SAP Mobile Platform.

Table 21. SiteMinder Single Sign-on Integration

Accessed Service	Details
SiteMinder-protected Web service	When an EIS Web service is protected by SiteMinder, SAP Mobile Platform sends the current SAP Mobile Platform user's SMSESSION cookie when executing the Web service. For more information on sending the SMSESSION cookie to the SiteMinder-protected Web service, see <i>Single Sign-on Using NamedCredential</i> in the <i>Security</i> guide.
SSO2-protected JCo RFC or SAP Web service	The SAP server is configured to use SSO2 tickets for single sign-on. SAP Mobile Platform sends the user's current MYSAPSSO2 ticket along with the request. SAP validates that the SSO2 ticket is valid and was issued by a trusted peer, and executes the request as that user. Note: You must have a SiteMinder agent for SAP installed in a NetWeaver server. SAP Mobile Platform sends the SMSESSION cookie to NetWeaver, the SAP SiteMinder agent validates the cookie, and then the TokenIssuingLoginModule generates an SSO2 ticket and returns it to SAP Mobile Platform as a MYSAPSSO2 cookie.
Web service hosted on NetWeaver requiring both SSO2 and SMSESSION	SAP Mobile Platform sends both SSO credentials when executing the Web service call.

Security Configuration to a SiteMinder-protected EIS

With SAP Mobile Platform, SiteMinder authentication is used in Network Edge and non-Network Edge configurations to authenticate the client of a Web service, SAP JCo, or NetWeaver service.

In your security configuration that integrates with SiteMinder applications, you need a ClientValuePropagatingLoginModule so you can save your SMSESSION cookie as a credential for EIS single sign-on. If the SiteMinder agent adds an sm_user header to client requests, use that header in the ClientValuePropagatingLoginModule to set a user Principal. If the SiteMinder agent does not add an sm_user header, then disable impersonation checking.

You should also have an `HttpAuthenticationLoginModule` configured for a SiteMinder-protected URL where SAP Mobile Platform can verify the validity of the user's `SMSESSION` cookie.

For a detailed example focusing on SiteMinder specific configurations for SAP Mobile Platform, see *How-To: Set up SUP with SiteMinder* at <http://scn.sap.com/docs/DOC-29574>.

Configuring Security for SiteMinder Token and Basic Authentication

Use SAP Control Center to create a security configuration for your single sign-on (SSO) applications.

1. In SAP Control Center, navigate to the **SAP Mobile Platform Cluster** pane and select **Security**.
2. In the **General** tab, click **New** and name your security configuration.
3. Open the **Security** folder and select your configuration. In the **Authentication** tab, click **Add** to add a LoginModule.
4. Choose the **ClientValuePropagatingLoginModule** and add these properties:
 - **Implementation Class** – `com.sybase.security.core.ClientValuePropagatingLoginModule`
 - **ClientHttpValuesAsPrincipals** – `sm_user`
 - **ClientHttpValuesAsNamedCredentials** – `smession:SMSESSION2`
 - **Control Flag**: optional

Note: `ClientHttpValuesAsNamedCredentials` ensures that if the client application picked up an `SMSESSION` cookie either using Network Edge authentication or an external token, it is saved as a credential named `SMSESSION2` on the subject so it can be used for SSO to a SiteMinder-protected EIS. Therefore, the `credential.a.name` property is `SESSION2`. Also, `ClientHttpValuesAsPrincipals` uses the `sm_user` HTTP header if the client has used Network Edge authentication and enables you to perform impersonation checking.

5. Click **OK**.
6. In the **Authentication** tab, select the default **NoSecLoginModule** and click **Delete**. LoginModule allows logins without credentials, and you must remove it for security integrity.
7. In the **Authentication** tab, click **New** to add a provider.
8. Select and configure the `HttpAuthenticationLoginModule`:
 - a) Select **`com.sybase.security.http.HttpAuthenticationLoginModule`** and click **Yes** in the Duplicate Authentication Provider warning.
 - b) Configure the module's properties so the SiteMinder-protected URL has the same policy server that issued the `SMSESSION` cookie to the client.
 - `ClientValuesToSend` = `SMSESSION`
 - `SendClientValuesAs` = `cookie:SMSESSION`

This causes SAP Mobile Platform to forward the cookie to the specified SiteMinder-protected URL. If the HTTP status response code is 200, then the SMSESSION cookie is valid and the user is considered authenticated.

9. In the **Authorization** tab, select the **NoSecAuthorizer** provider type and click **Delete**.
10. In the **Attribution** tab, select the **NoSecContributer** provider type and click **Delete**.
11. In the **Settings** tab, adjust the properties as follows:
 - **Authentication cache timeout(seconds)** – 0
 - **Maximum number of failed authentications** – 5
 - **Authentication lock duration(in seconds)** – 600
12. Click **Apply**.
13. In the **General** tab, click **Validate** to check your configuration.
14. With successful validation, click **Apply** to save all changes.

For detailed examples focusing on SiteMinder specific configurations for SAP Mobile Platform, see *How-To: Set up SUP with SiteMinder* at <http://scn.sap.com/docs/DOC-29574>.

Authentication Cache Timeout and Token Authentication

To reduce the load, SAP Mobile Platform uses an authentication cache to reduce the load it places on your back-end identity management and security systems. Depending on your security configuration, you can adjust the authentication cache timeout to avoid authentication failures and errors.

By default, the authentication cache holds a user's subject, principals, and credentials used for single sign-on to an EIS for 3600 seconds (one hour). If the user name and password contained inside subsequent SAP Mobile Platform requests are unchanged, the request is considered authenticated and uses the cached security information for access control and single sign-on to EIS operations.

Note: When using token-based authentication, clients should use a hash code of the token as the password, so SAP Mobile Platform proceeds through the login modules and replaces the cached token credential. This prevents using an expired token in single sign-on to an EIS.

If you cache an SMSESSION for a user and the token expires before the cache entry, you get authentication failures during the single sign-on EIS operations. This leads to either synchronization errors or operation replay errors.

Configure the authentication cache to avoid errors and failures. If needed, you can disable the authentication cache entirely by setting the cache timeout to 0. Every SAP Mobile Platform request is reauthenticated. For non-Network Edge basic authentication, you can set the cache interval to slightly less than the Idle Timeout for your SiteMinder session policy.

For Network Edge authentication, you must set the authentication cache timeout to 0. If the URL configured to validate the SMSESSION token also returns an HTTP header with the expiration time for the token expressed in milliseconds since the epoch (1/1/1970), the HttpAuthenticationLoginModule can use that value to adjust the authentication cache

expiration for this subject's entry so it expires at an appropriate time. Use the `TokenExpirationTimeHTTPHeader` to specify the name of the header containing this expiration value. Additionally, you can use `TokenExpirationInterval` property to reduce time from the expiration so it does not expire while SAP Mobile Platform is processing a request.

For detailed examples, including how to configure the timeout in the SiteMinder Admin, see *How-To: Set up SUP with SiteMinder* at <http://scn.sap.com/docs/DOC-29574>.

Configuring Network Edge Authentication with a SiteMinder-Protected Web Service

Configure the SMSESSION cookie for Network Edge authentication to a SiteMinder Web service.

Network Edge authentication for SiteMinder requires the Web service endpoint to be changed in SAP Control Center to use the SMSESSION cookie for single sign-on. By default, the application connection template is configured with the server name set to your SiteMinder-protected server.

1. In the left navigation pane, select **Cluster**, then expand **Domains**, and select the domain for which you want to create a new connection.
2. Select **Connections**.
3. In the right administration pane, select the **Connections** tab.
4. Select the EIS connection pool that is a Web service connection for the SiteMinder-protected service, and click **Properties**.
5. In the **Edit Connection Pool** pane, configure these properties:

Property	Value
credential.a.mapping	Cookie:SMSESSION
credential.a.name	SMSESSION

6. Click **Save**.

For detailed examples focusing on SiteMinder specific configurations for SAP Mobile Platform, see *How-To: Set up SUP with SiteMinder* at <http://scn.sap.com/docs/DOC-29574>.

Configuring Non-Network Edge Authentication with a SiteMinder-Protected Web Service

Configure the SMSESSION cookie and application connection template for non-Network Edge authentication to a SiteMinder-protected Web service.

Similar to Network Edge authentication for SiteMinder, non-Network Edge authentication requires the Web service endpoint to be changed in SAP Control Center to use the SMSESSION cookie for single sign-on. However, for non-Network Edge authentication, by default the application connection template is configured with the server name set to the SAP Mobile Platform server or a reverse proxy, depending on your configuration.

1. In the left navigation pane, select **Cluster**, then expand **Domains**, and select the domain for which you want to create a new connection.
2. Select **Connections**.
3. In the right administration pane, select the **Connections** tab.
4. Select the EIS connection pool that is a Web service connection pointing to the SiteMinder-protected service, and click **Properties**.
5. In the **Edit Connection Pool** pane, configure these properties:

Property	Value
credential.a.mapping	Cookie:SMSESSION
credential.a.name	SMSESSION

6. Click **Save**.

For detailed examples focusing on SiteMinder specific configurations for SAP Mobile Platform, see *How-To: Set up SUP with SiteMinder* at <http://scn.sap.com/docs/DOC-29574>.

Applications

An application consists of two parts: the software user interface that consumes enterprise data from one or more data sources, and the definition created in SAP Control Center that allows the mobile application to be recognized by the runtime. Depending on the application type, the application definition on the server can establish the relationship among packages used in the application, the domain that the application is deployed to, the activation code for the application, and other application specific settings.

SAP Mobile Platform supports multiple options for development and deployment of mobile applications.

- **Object API applications** – Object API applications are customized, full-featured mobile applications that use mobile data model packages, either using mobile business objects (MBOs) or Data Orchestration Engine, to facilitate connection with a variety of enterprise systems and leverage synchronization to support offline capabilities
- **Hybrid Apps** – HTML5/JS Hybrid Apps support simple business processes, such as approvals and requests, and also use mobile business objects (MBOs) to facilitate connection with a variety of enterprise systems. With this approach, a hybrid web container is developed and deployed to a device, then one or more workflows are deployed to the container.
- **OData applications** – OData applications are simple, mostly online mobile business applications that leverage standards-based approaches—OData protocol and RESTful Web Services.
- **Agency applications** – Agency applications provide the ability to support one or more client device types and one or more enterprise information systems of varying types from a single application project, high levels of configuration and modification with minimal user

impact, simple or complex multi-screen workflows, and high data volumes stored on client devices.

- **REST API applications** – Applications developed using 3rd Party SDKs, such as Sencha and PhoneGap, for any device platform, and using any programming language that supports sending and receiving HTTP messages can use the SAP Mobile Platform REST APIs to call services provided by the SAP Mobile Platform Runtime.

While developers create the client application using either SAP Mobile SDK or a third-party SDK, applications must be defined, then managed and monitored by administrators in SAP Control Center. The stages in the life cycle of an application may be different for different application types.

See also

- *Application and User Management Overview* on page 5
- *Considerations for Application Connection Registration* on page 322
- *Application ID and Template Guidelines* on page 320

Activating and Maintaining Applications

The activation stage defines all activities that relate to the startup of and formal entry of an application and a known or anonymous user into the SAP Mobile Platform runtime. It ties all entities together by associating a user with the application and its resources (connections, customizations, and packages).

The activation of applications is essential to their functionality. It is critical that you understand this activation process. SAP encourages you to read *The Activation Process* in *Mobile Application Life Cycle*. Once you understand what activation is and how it works, follow the administrator workflow in *Enabling Application Activation with SAP Control Center* of this same document.

Defining Applications

Applications are recognized by SAP Mobile Server by the properties that define them. Administrators define applications with a unique application ID and other key application properties, such as domain, packages, security configuration, and connection templates.

An application cannot register a connection unless a definition has been created for it. If your development team has not yet set these application properties, administrators must do so before the application connection can be registered.

Creating Object API Application Definitions

Create an Object API application definition by providing general application properties such as the application ID, description, security configuration, and domain details, and assigning MBO packages.

1. In the left navigation pane of SAP Control Center, click the **Applications** node and select the **Applications** tab in the right administration pane.
2. Click **New**.
3. Enter a unique Application ID.
 - SAP recommends that application IDs contain a minimum of two dots ("."). For example, the following ID is valid: `com.sap.mobile.app1`.
 - Application IDs cannot start with a dot ("."). For example, the following ID is invalid: `.com.sap.mobile.app1`.
 - Application IDs cannot have two consecutive dots ("."). For example, the following ID is invalid: `com..sap.mobile.app1`.
4. Enter a Display Name and Description for the application.
5. Select **Object API** as the Client SDK type.
6. Select the appropriate security configuration from the Security configuration drop-down list.
 For applications that do not require authentication, select the anonymous security configuration or the **Anonymous access** checkbox.
7. Select the appropriate domain from the Domain drop-down list.
8. Assign one or more MBO packages, as required.
9. Select a Base Template to use as the template for the application connection settings.
 To review the properties of the template, click **View**. You can modify the template properties after you create the application.
10. Click **OK** to complete the application definition.

See also

- *Creating Hybrid App Definitions* on page 258
- *Creating REST API Application Definitions* on page 260
- *Creating OData Application Definitions* on page 264
- *Creating Agency Application Definitions* on page 267
- *Native Notification Properties* on page 295
- *Application Customization Resource Bundles* on page 296

Configuring Native Notifications

(Does not apply to Agency SDK clients) Configure a new or edit an existing native notification properties for an application based on application ID and domain.

1. In the navigation pane of SAP Control Center, expand the **Applications** node and select the application.
2. Select the **Push Configurations** tab and click **Add**.
3. Configure the native configuration settings:

- a) Name - name of the configuration.
- b) Domain - the domain to which the configuration applies. A domain can have only one configuration for Apple (APNS) or Google (GCM) type notifications, but can have multiple Blackberry (BES) type configurations. .

Note: You must specify a domain when configuring push notifications for a Hybrid Web Container application.

- c) Type - the type of notification service determines the other configuration properties.
4. Configure native notification properties for the type of notification service you are enabling and click **OK**.
 5. Define the notification mode for the application.
 - To set the notification mode for a specific application connection template of the application, select the **Application Connection Templates** tab, and select the application connection template.
 - To set the notification mode for a specific application connection, select the **Application Connections** tab, and select the application connection.
- a) Select **Properties**.
 - b) Select **Application Settings**.
 - c) Select a value for **Notification Mode** that determines the way in which the notifications are delivered to the device:
 - Only native notifications - send notifications through native third party channels only (APNS for Apple devices, GCM for Android devices, BES for BlackBerry devices, and so on). Devices registered by the REST API only support native notification, because these applications do not have access to the client libraries required to receive a payload push notification.
 - Only Payload(Online) push - send reliable push messages (with business data as payload) through SAP Mobile Platform messaging channel only when the device is online. This mode is not supported for device registered by the REST API.
 - Payload(Online) push with native notification - send reliable push messages (with business data as payload) through SAP Mobile Platform messaging channel when the device is online, and use native notification through third party channels only if the device is off-line. This mode is not supported for device registered by the REST API.

See also

- *Native Notification Properties* on page 295

Changing Properties of Application Connection Templates

(Does not apply to Agentry SDK clients) Change existing property values used in application connection templates.

Note: Changes made to application connection template properties are applied to all application connections that use that template, except where the properties have been overridden for a specific application connection.

1. Do one of the following in the navigation pane of SAP Control Center:
 - Select the **Applications** node.
 - Expand the **Applications** node and select an application.
2. In the right administration pane, click the **Application Connection Templates** tab.
3. Select the application connection template from the list and click **Properties**.
4. In the Template dialog, select the category you want to edit and modify the property and value.

Note: Some properties are read-only and cannot be changed by the administrator. Some read-only properties are set by the application developer or the device. Other read-only properties are set by the administrator but cannot be changed after the template has been used to register an application connection.

5. Click **OK**.

Uploading Application Customization Resource Bundles

(Does not apply to Hybrid App or Agentry SDK clients) Before you can assign application customization resource bundles, you must upload them to SAP Mobile Server with SAP Control Center.

Only platform administrators can upload bundles to SAP Mobile Server.

1. In the navigation pane of SAP Control Center, expand the **Applications** node and select the application.
2. In the right administration pane, select the **Customization Resource Bundles** tab.
3. (Optional) Click **Refresh** to update the list of deployed customization resource bundles for this application.
4. (Optional) Add another customization resource bundle to the list.
 - a) Click **Add**.
 - b) In the file dialog, navigate to and select the customization resource bundle JAR file, and click **Open**. The name and version of the newly deployed customization resource bundle is added to the list.
 - c) (Optional) In the Confirm dialog, select one or more check boxes to assign the newly uploaded bundle to application connections or application connection templates with the same application ID. If no check boxes are selected, there is no automatic assignment.

Note: You can make these assignments at a later time.

- d) Click **OK**.
5. (Optional) Add another customization resource bundle to the application.
6. Click **OK**.

Creating Hybrid App Definitions

Create a Hybrid App definition by providing general application properties such as the application ID, description, security configuration and domain details, and assigning MBO and Hybrid App packages.

1. In the left navigation pane of SAP Control Center, click the **Applications** node and select the **Applications** tab in the right administration pane.
2. Click **New**.
3. Enter a unique Application ID.
 - SAP recommends that application IDs contain a minimum of two dots ("."). For example, the following ID is valid: `com.sap.mobile.appl`.
 - Application IDs cannot start with a dot ("."). For example, the following ID is invalid: `.com.sap.mobile.appl`.
 - Application IDs cannot have two consecutive dots ("."). For example, the following ID is invalid: `com..sap.mobile.appl`.
4. Select **Hybrid App** as the **Client SDK type**.
5. Select the appropriate security configuration from the Security configuration drop-down list.

For applications that do not require authentication, select the anonymous security configuration or the **Anonymous access** checkbox.
6. Select the appropriate domain from the Domain drop-down list.
7. Assign one or more MBO packages, as required.
8. Select a Base Template to use as the template for the application connection settings.

To review the properties of the template, click **View**. You can modify the template properties after you create the application.
9. Browse and select one or more Hybrid App packages to assign to the application connection template.
10. Click **OK** to complete the application definition.

See also

- *Creating Object API Application Definitions* on page 254
- *Creating REST API Application Definitions* on page 260
- *Creating OData Application Definitions* on page 264
- *Creating Agency Application Definitions* on page 267

- *Native Notification Properties* on page 295
- *Application Customization Resource Bundles* on page 296

Configuring Native Notifications

(Does not apply to Agentry SDK clients) Configure a new or edit an existing native notification properties for an application based on application ID and domain.

1. In the navigation pane of SAP Control Center, expand the **Applications** node and select the application.
2. Select the **Push Configurations** tab and click **Add**.
3. Configure the native configuration settings:
 - a) Name - name of the configuration.
 - b) Domain - the domain to which the configuration applies. A domain can have only one configuration for Apple (APNS) or Google (GCM) type notifications, but can have multiple Blackberry (BES) type configurations. .

Note: You must specify a domain when configuring push notifications for a Hybrid Web Container application.

- c) Type - the type of notification service determines the other configuration properties.
4. Configure native notification properties for the type of notification service you are enabling and click **OK**.
 5. Define the notification mode for the application.
 - To set the notification mode for a specific application connection template of the application, select the **Application Connection Templates** tab, and select the application connection template.
 - To set the notification mode for a specific application connection, select the **Application Connections** tab, and select the application connection.
 - a) Select **Properties**.
 - b) Select **Application Settings**.
 - c) Select a value for **Notification Mode** that determines the way in which the notifications are delivered to the device:
 - Only native notifications - send notifications through native third party channels only (APNS for Apple devices, GCM for Android devices, BES for BlackBerry devices, and so on). Devices registered by the REST API only support native notification, because these applications do not have access to the client libraries required to receive a payload push notification.
 - Only Payload(Online) push - send reliable push messages (with business data as payload) through SAP Mobile Platform messaging channel only when the device is online. This mode is not supported for device registered by the REST API.
 - Payload(Online) push with native notification - send reliable push messages (with business data as payload) through SAP Mobile Platform messaging channel when

the device is online, and use native notification through third party channels only if the device is off-line. This mode is not supported for device registered by the REST API.

Changing Properties of Application Connection Templates

(Does not apply to Agency SDK clients) Change existing property values used in application connection templates.

Note: Changes made to application connection template properties are applied to all application connections that use that template, except where the properties have been overridden for a specific application connection.

1. Do one of the following in the navigation pane of SAP Control Center:
 - Select the **Applications** node.
 - Expand the **Applications** node and select an application.
2. In the right administration pane, click the **Application Connection Templates** tab.
3. Select the application connection template from the list and click **Properties**.
4. In the Template dialog, select the category you want to edit and modify the property and value.

Note: Some properties are read-only and cannot be changed by the administrator. Some read-only properties are set by the application developer or the device. Other read-only properties are set by the administrator but cannot be changed after the template has been used to register an application connection.

5. Click **OK**.

Creating REST API Application Definitions

Create a REST API application definition by providing general application properties such as the application ID, description, security configuration and domain details, and defining the application and push endpoint.

1. In the left navigation pane of SAP Control Center, click the **Applications** node and select the **Applications** tab in the right administration pane.
2. Click **New**.
3. Enter a unique Application ID.
 - SAP recommends that application IDs contain a minimum of two dots ("."). For example, the following ID is valid: `com.sap.mobile.app1`.
 - Application IDs cannot start with a dot ("."). For example, the following ID is invalid: `.com.sap.mobile.app1`.
 - Application IDs cannot have two consecutive dots ("."). For example, the following ID is invalid: `com..sap.mobile.app1`.

4. Enter a Display Name and Description for the application.
5. Select **REST API** as the Client SDK type.
6. Select the appropriate security configuration from the Security configuration drop-down list.
For applications that do not require authentication, select the anonymous security configuration or the **Anonymous access** checkbox.
7. Select the appropriate domain from the Domain drop-down list.
8. Select a Base Template to use as the template for the application connection settings.
To review the properties of the template, click **View**. You can modify the template properties after you create the application.
9. Enter the URL pointing to the EIS as the **Application Endpoint**.
REST API applications require a proxy type connection endpoint. You can automatically create the proxy connection endpoint by entering a REST API URL as the Application Endpoint. This creates a proxy connection endpoint with the same name as the Application ID. If the REST API application uses an anonymous security configuration, the newly created connection endpoint will have the Allow Anonymous Access property set to True and the Address (URL) property set to the Application ID. If you want to create the proxy connection endpoint manually, create the proxy connection endpoint through the SAP Control Center Domains node.
10. Enter the SAP Mobile Platform URL used by the EIS to forward notifications as the **Push endpoint**,
11. Click **OK** to complete the application definition.

See also

- *Creating Object API Application Definitions* on page 254
- *Creating Hybrid App Definitions* on page 258
- *Creating OData Application Definitions* on page 264
- *Creating Agency Application Definitions* on page 267
- *Native Notification Properties* on page 295
- *Application Customization Resource Bundles* on page 296

Configuring Native Notifications

(Does not apply to Agency SDK clients) Configure a new or edit an existing native notification properties for an application based on application ID and domain.

1. In the navigation pane of SAP Control Center, expand the **Applications** node and select the application.
2. Select the **Push Configurations** tab and click **Add**.
3. Configure the native configuration settings:
 - a) Name - name of the configuration.

- b) Domain - the domain to which the configuration applies. A domain can have only one configuration for Apple (APNS) or Google (GCM) type notifications, but can have multiple Blackberry (BES) type configurations. .

Note: You must specify a domain when configuring push notifications for a Hybrid Web Container application.

- c) Type - the type of notification service determines the other configuration properties.
4. Configure native notification properties for the type of notification service you are enabling and click **OK**.
 5. Define the notification mode for the application.
 - To set the notification mode for a specific application connection template of the application, select the **Application Connection Templates** tab, and select the application connection template.
 - To set the notification mode for a specific application connection, select the **Application Connections** tab, and select the application connection.
- a) Select **Properties**.
 - b) Select **Application Settings**.
 - c) Select a value for **Notification Mode** that determines the way in which the notifications are delivered to the device:
 - Only native notifications - send notifications through native third party channels only (APNS for Apple devices, GCM for Android devices, BES for BlackBerry devices, and so on). Devices registered by the REST API only support native notification, because these applications do not have access to the client libraries required to receive a payload push notification.
 - Only Payload(Online) push - send reliable push messages (with business data as payload) through SAP Mobile Platform messaging channel only when the device is online. This mode is not supported for device registered by the REST API.
 - Payload(Online) push with native notification - send reliable push messages (with business data as payload) through SAP Mobile Platform messaging channel when the device is online, and use native notification through third party channels only if the device is off-line. This mode is not supported for device registered by the REST API.

Changing Properties of Application Connection Templates

(Does not apply to Agency SDK clients) Change existing property values used in application connection templates.

Note: Changes made to application connection template properties are applied to all application connections that use that template, except where the properties have been overridden for a specific application connection.

1. Do one of the following in the navigation pane of SAP Control Center:

- Select the **Applications** node.
 - Expand the **Applications** node and select an application.
2. In the right administration pane, click the **Application Connection Templates** tab.
 3. Select the application connection template from the list and click **Properties**.
 4. In the Template dialog, select the category you want to edit and modify the property and value.

Note: Some properties are read-only and cannot be changed by the administrator. Some read-only properties are set by the application developer or the device. Other read-only properties are set by the administrator but cannot be changed after the template has been used to register an application connection.

5. Click **OK**.

Uploading Application Customization Resource Bundles

(Does not apply to Hybrid App or Agentry SDK clients) Before you can assign application customization resource bundles, you must upload them to SAP Mobile Server with SAP Control Center.

Only platform administrators can upload bundles to SAP Mobile Server.

1. In the navigation pane of SAP Control Center, expand the **Applications** node and select the application.
2. In the right administration pane, select the **Customization Resource Bundles** tab.
3. (Optional) Click **Refresh** to update the list of deployed customization resource bundles for this application.
4. (Optional) Add another customization resource bundle to the list.
 - a) Click **Add**.
 - b) In the file dialog, navigate to and select the customization resource bundle JAR file, and click **Open**. The name and version of the newly deployed customization resource bundle is added to the list.
 - c) (Optional) In the Confirm dialog, select one or more check boxes to assign the newly uploaded bundle to application connections or application connection templates with the same application ID. If no check boxes are selected, there is no automatic assignment.

Note: You can make these assignments at a later time.

- d) Click **OK**.
5. (Optional) Add another customization resource bundle to the application.
6. Click **OK**.

Creating OData Application Definitions

Create an OData application definition by providing general application properties such as the application ID, description, security configuration and domain details, and defining the application and push endpoint.

1. In the left navigation pane of SAP Control Center, click the **Applications** node and select the **Applications** tab in the right administration pane.
2. Click **New**.
3. Enter a unique Application ID.
 - SAP recommends that application IDs contain a minimum of two dots ("."). For example, the following ID is valid: `com.sap.mobile.appl`.
 - Application IDs cannot start with a dot ("."). For example, the following ID is invalid: `.com.sap.mobile.appl`.
 - Application IDs cannot have two consecutive dots ("."). For example, the following ID is invalid: `com..sap.mobile.appl`.
4. Enter a Display Name and Description for the application.
5. Select **OData** as the **Client SDK type**.
6. Select the appropriate security configuration from the Security configuration drop-down list.

For applications that do not require authentication, select the anonymous security configuration or the **Anonymous access** checkbox.
7. Select the appropriate domain from the Domain drop-down list.
8. Select a Base Template to use as the template for the application connection settings.

To review the properties of the template, click **View**. You can modify the template properties after you create the application.
9. Enter the URL pointing to the EIS as the **Application Endpoint**.

OData applications require a proxy type connection endpoint. You can automatically create the proxy connection endpoint by entering an OData URL as the Application Endpoint. This creates a proxy connection endpoint with the same name as the Application ID. If the OData application uses an anonymous security configuration, the newly created connection endpoint will have the Allow Anonymous Access property set to True and the Address (URL) property set to the Application ID. If you want to create the proxy connection endpoint manually, create the proxy connection endpoint through the SAP Control Center Domains node.
10. Enter the SAP Mobile Platform URL used by the EIS to forward notifications as the **Push endpoint**.
11. Click **OK** to complete the application definition.

See also

- *Creating Object API Application Definitions* on page 254
- *Creating Hybrid App Definitions* on page 258
- *Creating REST API Application Definitions* on page 260
- *Creating Agency Application Definitions* on page 267
- *Native Notification Properties* on page 295
- *Application Customization Resource Bundles* on page 296

Configuring Native Notifications

(Does not apply to Agency SDK clients) Configure a new or edit an existing native notification properties for an application based on application ID and domain.

1. In the navigation pane of SAP Control Center, expand the **Applications** node and select the application.
2. Select the **Push Configurations** tab and click **Add**.
3. Configure the native configuration settings:
 - a) Name - name of the configuration.
 - b) Domain - the domain to which the configuration applies. A domain can have only one configuration for Apple (APNS) or Google (GCM) type notifications, but can have multiple Blackberry (BES) type configurations. .

Note: You must specify a domain when configuring push notifications for a Hybrid Web Container application.

- c) Type - the type of notification service determines the other configuration properties.
4. Configure native notification properties for the type of notification service you are enabling and click **OK**.
 5. Define the notification mode for the application.
 - To set the notification mode for a specific application connection template of the application, select the **Application Connection Templates** tab, and select the application connection template.
 - To set the notification mode for a specific application connection, select the **Application Connections** tab, and select the application connection.
 - a) Select **Properties**.
 - b) Select **Application Settings**.
 - c) Select a value for **Notification Mode** that determines the way in which the notifications are delivered to the device:
 - Only native notifications - send notifications through native third party channels only (APNS for Apple devices, GCM for Android devices, BES for BlackBerry devices, and so on). Devices registered by the REST API only support native

notification, because these applications do not have access to the client libraries required to receive a payload push notification.

- Only Payload(Online) push - send reliable push messages (with business data as payload) through SAP Mobile Platform messaging channel only when the device is online. This mode is not supported for device registered by the REST API.
- Payload(Online) push with native notification - send reliable push messages (with business data as payload) through SAP Mobile Platform messaging channel when the device is online, and use native notification through third party channels only if the device is off-line. This mode is not supported for device registered by the REST API.

Changing Properties of Application Connection Templates

(Does not apply to Agentry SDK clients) Change existing property values used in application connection templates.

Note: Changes made to application connection template properties are applied to all application connections that use that template, except where the properties have been overridden for a specific application connection.

1. Do one of the following in the navigation pane of SAP Control Center:
 - Select the **Applications** node.
 - Expand the **Applications** node and select an application.
2. In the right administration pane, click the **Application Connection Templates** tab.
3. Select the application connection template from the list and click **Properties**.
4. In the Template dialog, select the category you want to edit and modify the property and value.

Note: Some properties are read-only and cannot be changed by the administrator. Some read-only properties are set by the application developer or the device. Other read-only properties are set by the administrator but cannot be changed after the template has been used to register an application connection.

5. Click **OK**.

Uploading Application Customization Resource Bundles

(Does not apply to Hybrid App or Agentry SDK clients) Before you can assign application customization resource bundles, you must upload them to SAP Mobile Server with SAP Control Center.

Only platform administrators can upload bundles to SAP Mobile Server.

1. In the navigation pane of SAP Control Center, expand the **Applications** node and select the application.
2. In the right administration pane, select the **Customization Resource Bundles** tab.

3. (Optional) Click **Refresh** to update the list of deployed customization resource bundles for this application.
4. (Optional) Add another customization resource bundle to the list.
 - a) Click **Add**.
 - b) In the file dialog, navigate to and select the customization resource bundle JAR file, and click **Open**. The name and version of the newly deployed customization resource bundle is added to the list.
 - c) (Optional) In the Confirm dialog, select one or more check boxes to assign the newly uploaded bundle to application connections or application connection templates with the same application ID. If no check boxes are selected, there is no automatic assignment.

Note: You can make these assignments at a later time.

- d) Click **OK**.
5. (Optional) Add another customization resource bundle to the application.
6. Click **OK**.

Creating Agentry Application Definitions

Create an Agentry application definition by providing general application properties such as the application ID, description, security configuration and domain details. You can also deploy the application file to the SAP Mobile Server.

1. In the left navigation pane of SAP Control Center, click the **Applications** node and select the **Applications** tab in the right administration pane.
2. Click **New**.
3. Enter a unique Application ID.
 - SAP recommends that application IDs contain a minimum of two dots ("."). For example, the following ID is valid: `com.sap.mobile.app1`.
 - Application IDs cannot start with a dot ("."). For example, the following ID is invalid: `.com.sap.mobile.app1`.
 - Application IDs cannot have two consecutive dots ("."). For example, the following ID is invalid: `com..sap.mobile.app1`.
4. Select **Agentry** as the Client SDK type.
5. The Security configuration and Domain are preselected and cannot be changed.
 The Agentry security configuration is a predefined security configuration assigned to the "default" domain. It uses the `NoSecLoginModule`. This provider forwards login requests to the Agentry Server instance for authentication.
6. (Optional) Select **Application file** and browse to select the file. You can select:
 - **.agp files** – Agentry application definition files.

- **.aggpz files** – zip files containing Agentry application definition files and additional development artifacts.
 - **.zip files** – files that have been created from existing Agentry application definitions in SAP Control Center by downloading specific Agentry metadata.
7. If you are deploying the application file as part of application creation, specify the Active time for the application.
- You can activate the application immediately or at a specific date and time in the future. Selecting **Immediate** makes the application available to end users immediately, as long as the application is enabled.
8. Specify whether to **Enable after deployment**.
- Selecting this option automatically starts the Agentry Server instance for this application as soon as the application file is deployed (the Agentry Server instance status is `RUNNING`). If the application file has not been deployed, setting this option starts the Agentry Server instance without the application information (the Agentry Server instance status is `RUNNING WITHOUT METADATA`). If you do not select this option, you must manually enable the application before you can start the Agentry Server instance
9. Click **OK** to complete the application definition. Depending on the application definition:
- The application file is deployed.
 - The Agentry Server instance is started.
 - The application is available for end users.

Note: Any deploy errors or warnings are recorded in the SAP Mobile Server log. This log is located in `SMP_HOME\Servers\UnwiredServer\logs\<hostname>-server.log` or can be viewed in SAP Control Center under **Servers > primaryServer > Log**.

See also

- *Creating Object API Application Definitions* on page 254
- *Creating Hybrid App Definitions* on page 258
- *Creating REST API Application Definitions* on page 260
- *Creating OData Application Definitions* on page 264
- *Native Notification Properties* on page 295
- *Application Customization Resource Bundles* on page 296
- *Downloading Agentry Application Metadata for Deploying* on page 383

Configuring the Agentry Application Server Instance

Each Agentry application has its own Agentry Server instance that runs on the SAP Mobile Server node. You can modify the configuration of the Agentry Server instance, based on the

information loaded from the Agentry Server configuration file. The properties you can modify are dependent on the application definition.

Note: The Agentry Server instance for the application must be running in order for you to configure it. The Agentry Server instance status must be one of the following: `RUNNING`, `RUNNING WITHOUT METADATA` or `RUNNING NEED RESTART`.

1. In the navigation pane of SAP Control Center, expand the **Applications** node and select the application.
2. In the administration pane, click the **Configurations** tab.
3. Select a category and click **Properties**.
4. Modify the properties as required.

For details about configuring system connections, client-server communications, and localization settings, and for additional configuration tasks that require you to update configuration files directly, see *Agentry Server Administration* in *System Administration*.

If the configuration change requires a restart of the Agentry Server, a ! icon appears beside the property.

5. Click **Save**.

If any of the changes require a restart of the Agentry Server instance, the server instance status displays as `RUNNING NEED RESTART`. Click **Restart**.

Agentry Server Instance Configuration Categories and Properties

Each Agentry application has its own Agentry Server instance that runs on the SAP Mobile Server node. Agentry Server instance configuration properties are grouped by category.

Agentry Server Configuration Sections

You can access configuration settings for Agentry Server in SAP Control Center.

The following table describes the general purpose and areas of behavior affected by the settings contained in each section. These settings are ultimately contained within the `Agentry.ini` configuration file for the Agentry Server. The preferred method of setting any configuration option is to use SAP Control Center wherever possible. However, some settings may need to be initially set within the `Agentry.ini` file. In such cases these settings are noted in their descriptions, as well as in any procedures which describe modifying those settings.

Section	Description
Agentry	Contains settings related to the messages and events logs, as well as one to identify the Server instance.

Section	Description
Server	Contains settings for user, device, and expiration keys. These may be changed if new licenses are provided by Syclo in relation to changes in the usage of the platform. Also contains general sever configuration options related to time outs.
Configuration	This section contains settings for referencing other configuration files used by the Server. Also, the settings specifying if the Server is configured to provide debugging and whether the Server is for development or production uses.
Front Ends	This sections lists the dynamic link libraries (DLL's) to load for the supported Client-Server communications methods. This list is modified if different connection types are supported other than ANGEL.
ANGEL Front End	The configuration options in this section affect the behavior of communications between the Clients and Server when using the ANGEL connection type. It includes certificate file references, client and server authentication settings, cipher strengths for encryption, and time out values.
ANGEL Front End Ports	This section contains a list of one or more host names and ports upon which the Server will listen for requests from Clients. Multiple entries may be added here for environment support such as Network Address Translation, security and firewalls, and clustered environments.
Web Server Front End	This section contains configuration options that affect the behavior of communications between the Clients and Server when the Clients are web browsers. This includes both PC browsers and the RIM Blackberry Hand-held Browser. Options related to authentication certificates, ports, and debugging for such communications are set here.
Web Server MIME Types	This section lists the MIME types related to Web Clients. The items listed here are in addition to those configured on the systems in use.
Server Side Clients	This section lists the DLL files used for Web Server communications. Currently only the entry for ServerSideClient.dll should be listed here. Other options within this section are reserved for future use.
System Connections	This section lists the DLL files to be loaded by the Server based on the system connections the application supports and uses. Each entry in this section must match a System Connection definition within the application. Additionally, a matching section must exists for the System Connection within the Agentry.ini file.

Section	Description
SQL-n	This section contains the configuration options for a single SQL Database system connection. Its settings control the behavior of this connection between the Server and a database system. There may be multiple SQL-n sections within the Agentry.ini file. One must exist for each SQL Database system connection definition within the application. The n portion of the section name must be replaced with the ID value generated by the Editor for the system connection definition.
Java-n	This section contains the configuration options for a single Java Virtual Machine system connection. Its settings control the behavior of this connection between the Server and a Java interface. There may be multiple Java-n sections within the agentryr.ini file. One must exist for each Java Virtual Machine system connection definition within the application. The n portion of the section name must be replaced with the ID value generated by the Editor for the system connection definition.
HTTPXML-n	This section contains the configuration options for a single HTTP-XML system connection. Its settings control the behavior of this connection between the Server and an HTTP server. There may be multiple HTTPXML-n sections within the Agentry.ini file. One must exist for each HTTP-XML system connection definition within the application. The n portion of the section name must be replaced with the ID value generated by the Editor for the system connection definition.
File-n	This section contains the configuration options for a single NT File system connection. Its settings control the behavior of this connection between the Server and the Server's host system. There may be only one File-n section within the Agentry.ini file. This section must exist for an NT File system connection definition within the application. The n portion of the section name must be replaced with the ID value generated by the Editor for the system connection definition.
Agents	This section is used by the Server for internal purposes. It should not be modified unless specifically instructed by a Syclo support or services associate.
SpinDoc	This section is used by the Server to determine the location of SQL script files. It should not be modified unless specifically instructed by a Syclo support or services associate.
LastUpdates	This section contains date and time values related to the last update of certain other sections within the Server's configuration files. It should never be modified and is used by the Server for internal purposes.

Agentry Server: [Agentry] Configuration Section

The following table lists the available configuration options, acceptable values, and descriptions of the settings in the Agentry set of properties found in the SAP Control Center. These settings are also those found in the [Agentry] section of the `Agentry.ini` file.

Configuration Setting	Acceptable Values	Description
systemName	Any Text Value	This optional parameter specifies the name of the Agentry Server. This value is displayed in the title bar of the Server's GUI interface on the console. If no value is specified, or if the setting is omitted from the section, the default value of "Agentry Server" is displayed.
messageLogFile	Valid file name or path and file name. Default: messages.log	This parameter contains the name of the file to which message log items are written. In most cases the default of messages.log is sufficient. It can be changed to specify a different file name or location. The path can be relative to the Server's installation folder; or a full path including drive letter.
eventLogFile	Valid file name or path and file name. Default: events.log	This parameter contains the name of the file to which event log items are written. In most cases the default of events.log is sufficient. It can be changed to specify a different file name or location. The path can be relative to the Server's installation folder; or a full path including drive letter.
logMaxSize	Numeric value, assumed to be in bytes. Default: 0	The maximum size of the messages.log file in bytes. A value of 0 indicates no maximum size. Any other value will result in the log file roll behavior when the messages.log file exceeds the value set here.
logRollTime	Time of day in 24 hour clock format. Hour and minute of the hour only. Default: 1:45	The time of day, in 24 hour format (HH:MM) when the events and messages log files should be rolled. The Server must be running at the time specified here in order for the files to be rolled.
logRollCommand	Valid batch file name or path and batch file name. Default: roll.bat	The name of the batch file or command to roll the events and messages log files. The default roll.bat is installed with the Server. The command referenced here is executed when the log roll time (logRollTime) is reached, or the maximum log size (logMaxSize) is exceeded.

Configuration Setting	Acceptable Values	Description
logRollsAtTime	Boolean value of “true” or “false.” Default: true	Boolean value of true or false indicating whether to roll the messages and events log files based on the time in logRollTime.

Agency Server: [Server] Configuration Section

The following table lists the available configuration options, acceptable values, and descriptions of the settings in the Server set of properties found in the SAP Control Center. These settings are also those found in the [Server] section of the `Agency.ini` file.

Configuration Setting	Acceptable Values	Description
userKey	Valid Syclo-issued user key	Initially this will contain the user key entered when the Server installer was run. It can be modified with an updated user key, issued by Syclo, if additional user licences are purchased.
deviceKey	Valid Syclo-issued device key	Initially this will contain the device key entered when the Server installer was run. It can be modified with an updated user key, issued by Syclo, if additional device licenses are purchased.
expKey	Valid Syclo-issued expiration key	Initially this contain the expiration key entered when the Server installer was run. It can be modified with an updated expiration key, issued by Syclo, if a new expiration date is set.
autoRegister	Boolean of true or false. Default: true	This configuration option controls whether or not user ID's sent by the Client to the Server are added to the users.ini file automatically. If this option is set to false, the users.ini file will not be updated by the Server. Any user ID not listed in the users.ini file will not be allowed to log into the Server.
inactiveTimeout	Numerical value representing seconds. Default: 600	This is a duration value specifying how long the Server is to wait for response from a Client after the Server has sent it a message. If no response is received from the Client after this duration value, the user is logged out of the Server. This setting affects all Client communication types.

Configuration Setting	Acceptable Values	Description
allowRelogin	Boolean of true or false. Default: true	This setting controls whether users can log into the Server from Clients with different IP addresses at the same time. If true, the original session for a user will be ended, with the user logged out, before they are logged in from the new IP address. If this option is false, the Server will reject any Client login attempts from a second IP address for a user currently logged in from a different address.

Note: Two additional options, `notifyUserOnLogout` and `notifyUserOnShutdown` may be listed in the [Server] section of the `Agentry.ini` file. These items are no longer supported and are scheduled to be deprecated in a future release of `Agentry`.

Agentry Server: [Configuration] Configuration Section

The following table lists the available configuration options, acceptable values, and descriptions of the settings in the Configuration set of properties found in the SAP Control Center. These settings are also those found in the [Configuration] section of the `Agentry.ini` file.

Configuration Setting	Acceptable Values	Description
<code>developmentServer</code>	Boolean value of “true” or “false.” Default: <i>See Description.</i>	This configuration option controls whether the Server behaves as a development or production server. This option is set by the installer based on the options selected there (development: true; production: false) and should not be changed except at the instruction of a Syclo support or services associate.
<code>debug</code>	Boolean value of “true” or “false.” Default: <i>See Description.</i>	This configuration option controls whether certain log files are generated by the Server at run-time. It’s default values are true for a development server and false for production. Log Files: <code>Server.log</code> , <code>Server.user.log</code> , <code>AgentryJava.log</code> .

Configuration Setting	Acceptable Values	Description
localizations	Text value of valid locale names. Default: <i>none</i>	This configuration setting can contain one or more local names according to the ISO 639-1 standard. This option lists the supported languages, requiring corresponding override localization files, for an application. Each local name must be separated by a semi-colon.
localizationPath	Valid path relative to the Server's installation folder. Default: localizations	This option contains the location of localization files for use in the application. This path is relative to the Server's installation folder. If the default folder "localizations" does not exist, it must be created.
enableOverrideFile	Valid file name or path and file name. Default: Enables.ini	This configuration option specifies the name and location of the enables localization file. This setting can be changed when the name of this file is different from the default.
clientStringsFile	Valid file name or path and file name. Default: ClientText.ini	This configuration option specifies the name of the client strings localization file. This setting can be changed when the name of this file is different from the default.
applicationStringsFile	Valid file name or path and file name. Default: ApplicationText.ini	This configuration option specifies the name and location of the application strings localization file. This setting can be changed when the name of the file is different than the default.
applicationGlobalsFile	Valid file name or path and file name. Default: Globals.ini	This configuration option specifies the name and location of the globals localization file. This setting can be changed when the name of the file is different than the default.
clientStringNames	Valid file name or path and file name. Default: ClientStringNames.ini	This configuration option specifies the name and location of the client string names localization file. This setting can be changed when the name of the file is different than the default.

Configuration Setting	Acceptable Values	Description
overrideTypesFile	Valid file name or path and file name. Default: Override-Types.ini	This configuration option specifies the name and location of the override types localization file. This setting can be changed when the name of the file is different than the default.
transmitConfigurationFile	Valid file name or path and file name. Default: TransmitConfigurations.ini	This configuration option specifies the name and location of the transmit configurations override file. This setting can be changed when the name of the file is different than the default.
enableTransactionFailureHandling	Boolean value of "true" or "false." Default: false	This configuration option enables or disables the transaction failure handling functionality within the Agentry platform. This option must be true in order for all aspects of this functionality to be enabled.
enableFailedTransactionLogging	Boolean value of "true" or "false." Default: true	This configuration option enables or disables the failed transaction que, which is a part of the transaction failure handling functionality. By default, when transaction failure handling is enabled, transactions that fail with a fatal error code result in the creation of a failed transaction que on the server containing the data from that transaction. If this option is set to false, this file is not generated. All remaining transaction failure handling behaviors remain the same, including the removal of the failed transaction from the client. NOTE: setting this option to false can result in lost data for fatal errors related to transaction processing. This option has no affect if transaction failure handling is disabled, i.e enable-TransactionFailureHandling is false.

Configuration Setting	Acceptable Values	Description
failedTransactionQueue	Valid folder name or path and folder name. Default: FailedTransactionsQueue	This configuration option specifies the name of the folder where the failed transaction queue files are stored. This path is relative to the installation folder of the Server. It may be changed if the default folder is not the desired location for these files.
failedTransactionFilename-Format	Series of format strings and constants producing a valid XML file name.	This configuration option specifies the format of the file name for each failed transaction queue file generated by the Server. The format strings <code>%{userid}</code> , <code>%{transaction_name}</code> , <code>%{date}</code> , <code>%{time}</code> , and <code>%{count}</code> may be used. The file generated will always be in XML format and the file extension provided (.xml) should reflect this content type.

Agentry Server: [Front Ends] Configuration Section

The following table lists the available configuration options, acceptable values, and descriptions of the settings in the Front Ends set of properties found in the SAP Control Center. These settings are also those found in the `[Front Ends]` section of the `Agentry.ini` file.

The items in this section are anonymous keys. Therefore, the following table lists the acceptable values, specifying library (DLL) files to be loaded by the Server upon startup in support of the different client-server connection types. Only those for front ends (connect types) in use should be listed in this section.

Library File Name	Description
AngelVine.dll	This file is listed in the <code>[Front Ends]</code> section by default for versions of Agentry beginning with 4.4. This file must be listed if one or more transmit configurations are defined to use the ANGEL connection type.
WebVine.dll	This file is listed in the <code>[Front Ends]</code> section when the Web Client is in use for an application implementation. When listed, the Server will take on certain web server-like behaviors, serving screens in HTML or WML format to web clients.

Agentry Server: [ANGEL Front End] Configuration Section

The following table lists the available configuration options, acceptable values, and descriptions of the settings in the ANGEL Front End set of properties found in the SAP Control Center. These settings are also those found in the [ANGEL Front End] section of the `Agentry.ini` file. If the ANGEL connect type is not in use by at least one transmit configuration definition within the application, the settings of this section have no impact on the behavior of the Agentry Server.

Configuration Setting	Acceptable Values	Description
trustedCertificateStore	Valid Trusted Certificate Store File Name. Default: none	This option specifies the location of the trusted certificate store for the Server. This store is only used when Clients are required to provide authentication certificates during synchronization. Proper format for trustedCertificateStore is a PEM file that contains one or more trusted certificates.
authenticationCertificateStore	Valid authentication certificate file.	This option specifies the location of the authentication certificate for the Server. Only Personal Information Exchange formatted files (.pfx) are supported.
authenticationCertificateStorePassword	Valid password for accessing the Server's authentication certificate.	This option specifies the password to access the Server's authentication certificate.
authenticationCertificateStorePasswordEncoded	Boolean value of "true" or "false." Default: false	This option specifies whether or not the certificate store password value is encoded in the <code>Agentry.ini</code> file. This option should not be changed manually. Rather, the quick password encoder utility will change it when run.

Configuration Setting	Acceptable Values	Description
authenticateClient	Boolean value of “true” or “false.” Default: false	This option controls whether or not Clients are required to provide an authentication certificate to the Server during synchronization. This certificate must be listed in the trusted certificate store in order for Clients to access the Server.
debug	Boolean value of “true” or “false.” Default: false.	This configuration file controls whether or not the Server generates log files for communications using the ANGEL protocol. When true, the log file listed in debugFile is generated, as are individual log files for each Client during synchronization.
debugFile	Valid log file name or path name name. Default: ANGEL.log	This configuration option specifies the name of the log file generated by the Server when the debug option in this same section is true.
timeout	Duration value in seconds. Default: 300	This configuration option specifies how long the Server will keep a connection option with a Client when no messages or responses are received from the Client.
keepAliveTime	Duration value in seconds. Default: 60	This configuration option specifies how often the Server will send a keep alive message to the Client when the Client is keeping an open connection (Push or Background Sending) with the Server. This message prevents the connection from expiring based on the timeout value.

Agency Server: [Web Server Front End] Configuration Section

The following table lists the available configuration options, acceptable values, and descriptions of the settings in the Agency set of properties found in the SAP Control Center. These settings are also those found in the [Web Server Front End] section of the

Agency.ini file. The settings in this section apply only to Web Client functionality. They will have no affect on the Agency Server's behavior if this client type is not in use.

Configuration Setting	Acceptable Values	Description
listenOn	Valid port number. Default: see description	This setting specifies the port number upon which the Server will listen for requests from web browser clients. The default for unsecured traffic is port 80 and 443 for SSL encrypted traffic. This may be changed to any valid and open port number. It should not be the same as any other port number used by the Server for other client-server communications.
permitGetCGI	Boolean value of "true" or "false." Default: true	This setting specifies whether or not the Server should accept GET CGI requests from web browser clients.
debug	Boolean value of "true" or "false." Default: false	This setting controls whether or not the log file for web client communications is generated.
xslDirectory	Valid folder or path. Default: xsl	This setting contains the name of the folder where the XSL files served to browser clients are stored. This is relative to the Server's installation folder.
resourceDirectory	Valid folder or path. Default: resource	This setting contains the location of the folder where the resources for the web pages served to the browser clients are stored. This location is relative to the Server's installation folder.
authenticationCertificateStore	Valid location of authentication certificate.	This setting references the authentication certificate provided to browser clients by the Server. The format of the file given to authenticationCertificateStore should be PKCS#12.

Configuration Setting	Acceptable Values	Description
authenticationCertificateStorePassword	Valid password for authentication certificate.	This setting is the password value needed to access the authentication certificate referenced in authenticationCertificateStore. This value is changed when a different certificate is used that has a different password value.
authenticationCertificateStorePasswordEncoded	Boolean value of “true” or “false.” Default: false	This setting specifies whether or not the password referenced in authenticationCertificateStorePassword has been encoded by the quick password encoder utility. This setting should not be changed manually. It will be set to the proper value by the encoder utility when run.

Agentry Server: [Web Server MIME Types] Section

The [Web Server MIME Types] section of the `Agentry.ini` file is used to provide the proper Multipurpose Internet Mail Extensions (MIME) types settings for an application using the Web Server front end. The entries listed in this section are MIME types to be supported in addition to those registered on the host system of the Agentry Server.

The entries for this section take the form:

```
fileExtension=MIMETYPE
```

If a MIME type is listed in the [Web Server MIME Types] section of the `Agentry.ini` file and also registered on the Windows host system of the Server, the entry in the `Agentry.ini` file will override the registered entry.

Agentry Server: [System Connections] Section

The following table lists the available configuration options, acceptable values, and descriptions of the settings in the System Connections set of properties found in the SAP Control Center. These settings are also those found in the [System Connections] section of the `Agentry.ini` file.

The items listed in this section are key and values, where the key portion is the same as the ID of the system connection definition for which the entry is added. Therefore, the following table lists the acceptable values, specifying library (DLL) files to be loaded by the Server upon startup in support of the different system connection types supported by the Agentry platform. This section should contain entries only for system connection in use by the application and

defined within the application project. A corresponding system connection type section ([SQL-*n*], [Java-*n*], [HTTPXML-*n*], or [File-*n*]) must be added to the `Agentry.ini` file as well.

Note: The items in this section are typically added automatically by the Agentry Editor when needed and based on the system connections defined within the project when that project is published to the Agentry Server. Therefore it is rarely necessary to add these items manually. They are listed here for reference purposes only.

Library File Name	Description
ag3sqlbe.dll	This file is listed for each SQL Database system connection within the application.
ag3javabe.dll	This file is listed for each Java Virtual Machine system connection within the application.
ag3httpxml.dll	This file is listed for each HTTP-XML system connection within the application.
ag3filebe.dll	This file is listed when an NT File System connection is in use within the application.

*Agentry Server: [SQL-*n*] Section*

The following table lists the available configuration options, acceptable values, and descriptions of the settings in the SQL-*n* set of properties found in the SAP Control Center. These settings are also those found in the [SQL-*n*] section of the `Agentry.ini` file.

When adding such a section the *n* portion of the section name must match the ID of the corresponding SQL Database system connection definition in the application. If no such system connection exists, the settings in this section will not affect the behavior of the Agentry Server or the application.

Configuration Setting	Acceptable Values	Description
name	Text value. Default: <i>null</i>	This setting can contain any text value used to identify the system connection in log files and other areas. This should be set to a unique value especially when working with multiple system connections.

Configuration Setting	Acceptable Values	Description
sharedConnections	Zero or positive integer value. Default: 0	This setting specifies the number of connections created by the Server upon startup. If greater than 0, a connection pool is created and data from a single Client may be processed on any of these connections, including multiple different connections during a single transmission. If set to 0, each Client transmission will use a single database connection from start to finish.
queryInitFile	Valid file or path and file name to properly formatted query initialization file. Default: sqlBE.ini	This setting specifies the name and location for the query initialization file used by this system connection. This initialization file contains numerous sections related to various Server events when working with a database system connection. The file referenced here is expected to contain all of these sections in the proper format.
debug	Boolean value of “true” or “false.” Default: <i>see description</i> .	This setting specifies whether or not the log file referenced in queryDebugFile is generated by the Server. It also enables or disables the creation of the sql.query.user.log file. This setting is false by default for production servers and true for development servers.
queryDebugFile	Valid file or path and file name. Default: sql.query.log	This setting specifies the log file generated by the server to contain log messages for the system connection. This includes query results and database generated messages. This file name may be changed when multiple database system connections are in use. The name may include the token %1 to include the text value from the name configuration option in the log file’s name. Example: “%1.sql.query.log” would produce a log file named similar to: “MyDB.sql.query.log”

Configuration Setting	Acceptable Values	Description
saveQuery	Boolean value of “true” or “false.” Default: <i>see description</i>	This setting specifies whether or not the query being executed is included in the query debug log file. This setting is false by default for production servers and true for development servers.
saveResult	Boolean value of “true” or “false.” Default: <i>see description</i>	This setting specifies whether or no the resulting messaging from the database is included in the query debug log file. NOTE: Error messages from the database are always written to the log file. This setting is false by default for production servers and true for development servers.
queryConstantsFile	Valid file name or path and file name. Default: <i>see description</i>	This setting specifies the file name and location, relative to the Server, of the query constants file. The default for this setting depends on the type of database selected during installation and will be set to either oracle_sd.ini or sqlserver_sd.ini.
preloadQueryInitFile	Boolean value of “true” or “false.” Default: <i>see description</i>	This setting specifies whether query initialization file named in the queryInitFile setting is loaded when the Server starts up, or if it is read by the Server for each new Client connection. If false, changes to the query initialization file will not take affect until the Server is restarted. This setting is false by default for production servers and true for development servers.
enableAuthentication	Boolean value of “true” or “false.” Default: true	This value specifies whether or not users are authenticated against the back end system for this system connection. At least one system connection within the application must perform user authentication.
enablePreviousUserAuthentication	Boolean value of “true” or “false.” Default: true	This value specifies whether or not previous users are authenticated against the back end system for this system connection. This authentication occurs when a user change occurs on the Client.

Configuration Setting	Acceptable Values	Description
useUserDBAuthConnection	Boolean value of “true” or “false.” Default: false	This value specifies whether or not the connection used to authenticate the user should be the same one used to process synchronization for that user. If false, the Server will use a different connection.
allowUserToChangePassword	Boolean value of “true” or “false.” Default: true	This value specifies whether or not users can change their passwords on the Client when the database indicates the password has or is about to expire. If enableAuthentication is false, this setting has no affect.
dbConnectionName	Text value with valid system connection name. Default: <i>set by installer.</i>	The value of this setting contains the TNS name or ODBC DSN name the Server will use to connect with the database.
dbConnectionUserID	Text value with valid user ID	The user ID the Server will use to log into the database.
dbConnectionPassword	Text value with valid password. Default: <i>set by installer.</i>	The password for the user ID the Server will use to log into the database.
dbConnectionPasswordEncoded	Boolean value of “true” or “false.” Default: false	Specifies whether or not the value in dbConnectionPassword has been encoded by the quick password encoder utility. This value should not be changed manually as the encoder utility will set it appropriately.
dbConnectionType	“Oracle” or “ODBC.” Default: <i>set by installer.</i>	Specifies whether the Server is connecting to the database using an ODBC or Oracle Net Service Name.
disconnectFromDB	Boolean value of “true” or “false.” Default: false	This Boolean value specifies whether or not the Server should disconnection from the database at the time specified in disconnectTime. This is set to true when the database performs periodic batch or other processing and when such processing dictates that no connections should be made to the database system.

Configuration Setting	Acceptable Values	Description
disconnectTime	24 hour clock value in hours and minutes. Default: 2:30	This setting contains the time of day when the Server should disconnect from the database. This setting has no affect if disconnectFromDB is false.
reconnectTime	24 hour click value in hours and minutes. Default: 4:00	This setting contains the time of day when the Server should reconnect to the database. It must be later than the time specified in disconnectTime. This setting has no affect if disconnectFromDB is false.
timeZoneName	Valid time zone name.	This setting contains the name of the time zone to which date and time values are converted when received from Clients; or from which they are converted when sent to Clients. This conversion is based on the application definitions. This setting is only used to specify a time zone other than the one for the back end system.

Agentry Server: [Java-n] Section

The following table lists the available configuration options, acceptable values, and descriptions of the settings in the Java-*n* set of properties found in the SAP Control Center. These settings are also those found in the [Java-*n*] section of the `Agentry.ini` file.

When adding such a section the *n* portion of the section name must match the ID of the corresponding Java Virtual Machine system connection definition in the application. If no such system connection exists, the settings in this section will not affect the behavior of the Server or the application.

Configuration Setting	Acceptable Values	Description
enableAuthentication	Boolean value of "true" or "false." Default: true	This value specifies whether or not users are authenticated against the back end system for this system connection. At least one system connection within the application must perform user authentication.
enablePreviousUserAuthentication	Boolean value of "true" or "false." Default: true.	This value specifies whether or not previous users are authenticated against the back end system for this system connection. This authentication occurs when a user change occurs on the Client.

Configuration Setting	Acceptable Values	Description
name	Text value	This setting can contain any text value used to identify the system connection in log files and other areas. This should be set to a unique value especially when working with multiple system connections.
classPath	Valid fully qualified paths.	This setting contains multiple path values specifying the location of different Java resources. This can include .jar files used by the application. Each path is required to contain be a fully qualified path. Each entry must be separated by a semi-colon. All paths for this setting are added to the system variable CLASSPATH of the Server's host system.
serverClass	Valid Java Server Class extension	This option is set to specify the application-specific extension of the AJ-API syclo.agentry.server class. This setting is set when this API class has been extended by a server class for the application. If no such class exists, the default syclo.agentry.server class is assumed and should not be set for this configuration option.
outputDirectory	Valid path value, relative to the Server.	This configuration option is set to the location where the compiled .class files produced by the JVM are to be stored for execution. The standard is to use the sub-folder Java from the Server's installation folder. However, another location may be used. Whichever location is used it must be specified here. This same path value must also be a part of the system's CLASSPATH. This can be accomplished by setting that system variable, or by adding the value to the classPath configuration option.

Configuration Setting	Acceptable Values	Description
sourceDirectory	Valid path value, relative to the Server.	This configuration option specifies the location of the .java files for the Steplet, ComplexTable and DataTable classes. This location is where these files are placed prior to compilation by the JVM. Note that this is not the location where the files reside in any project for Java files related to the application. The .java files are written to this location by the Server at run-time. Their contents are based on the definitions in which they are contained.
deleteSource	Boolean value of “true” or “false.” Default: false	This configuration option controls whether the files stored in sourceDirectory are deleted after successfully compiled by the JVM. This may be necessary in a development environment if the Java source files are modified outside of the Editor.
printStackTrace	Boolean value of “true” or “false.” Default: false	Setting this option to true will result in messages generated by the exception stack trace being printed to the events.log file produced by the Server.
printBusinessLogicStackTrace	Boolean value of “true” or “false.” Default: false.	Setting this option to true will result in messages generated by the AJ-API exception class JavaBusinessLogicError being printed to the events.log file produced by the Server.
timeZoneName	Valid time zone name. Default: null	This setting contains the name of the time zone to which date and time values are converted when received from Clients; or from which they are converted when sent to Clients. This conversion is based on the application definitions. This setting is only used to specify a time zone other than the one for the back end system.
initialHeapSize	Positive integer value in megabytes.	This setting specifies the initial heap size allocated by the JVM when it is started by the Server. This is equivalent to passing the -Xms command-line option to the JVM.

Configuration Setting	Acceptable Values	Description
maxHeapSize	Positive integer value in megabytes.	This setting specifies the max heap size the JVM will allocate when running. This is the equivalent to passing the -Xmx command-line option to the JVM
reduceOSSignalUse	Boolean value of “true” or “false.” Default: false	This option, when set to true, will result in the JVM not being shut down when the Server shuts down. This is only applicable when the Server is running as Windows service.
nonStandardJavaOptions	Text value.	This option can contain additional command-line options passed to the JVM when started by the Server. Any valid options to the JVM may be listed here as they would appear on the command-line. If either -Xms or -Xmx are specified here, the corresponding configuration settings should not be used and should be omitted from the [Java-n] section.
lowMemoryPercentage	Positive integer value as a percentage.	This configuration option can contain a numeric value treated as a percentage. If available system memory falls below this value a message is written to the events.log file generated by the Server.
performCompile	Boolean value of “true” or “false.” Default: true.	This configuration option specifies whether or not to compile the .java files into .class files for the Steplet, ComplexTable, and DataTable classes written for the application. Setting this to false may increase performance in a production environment. In this case, the last built .class files will be reused each time such a class is called within the application. If set to true, each class of the above types will be recompiled prior to execution for each user.
debugFile	Valid file name or path and file name relative to the Server.	This option specifies the name of the log file generated by the Server with messages produced by the AJ-API. The token %1 may be included to make add the value in the “name” configuration option of the [Java-n] section to the file name. Example: “%1.java.log” will produce a file name similar to: “MyJVM.java.log”

Configuration Setting	Acceptable Values	Description
debug	Boolean value of “true” or “false.” Default: <i>see description</i>	This option specifies whether debugging messages are generated by the Server in relation to the JVM System Connection for the [Java- <i>n</i>] section. Note that this option need not be set to true in order to have the JVM produce its own debugging information via the nonStandardJavaOptions setting. The debug option is set by default to true for development servers and false for production servers.

Agentry Server: [HTTPXML-*n*] Section

The following table lists the available configuration options, acceptable values, and descriptions of the settings in the HTTPXML-*n* set of properties found in the SAP Control Center. These settings are also those found in the [HTTPXML-*n*] section of the Agentry.ini file.

When adding such a section the *n* portion of the section name must match the ID of the corresponding HTTP-XML system connection definition in the application. If no such system connection exists, the settings in this section will not affect the behavior of the Server or the application.

Configuration Settings	Acceptable Values	Description
name	Text Value	This setting can contain any text value used to identify the system connection in log files and other areas. This should be set to a unique value especially when working with multiple system connections.
baseURL	Valid URL and optional Port number. Default: <i>none</i>	This is the base URL address of HTTP requests made by the Server for this system connection. This option must be set prior to starting the Server.
xmlNamespaces	Valid XML namespaces within XML schema	This configuration option can contain one or more XML name spaces within the XML schema. Each name space entry for this option must be separated by a pipe () symbol.

Configuration Settings	Acceptable Values	Description
debugFile	Valid file name or path and file name relative to Server folder. Default: httpxml.log	This option specifies the name of the log file generated by the Server. The token %1 may be included to add the value in the “name” configuration option of the [HTTPXML-n] section to the file name. Example: “%1.httpxml.log” will produce a file name similar to: “MyXML.httpxml.log”
user Debug File	Valid file name or path and file name relative to Server folder. Default: httpxml.%1.log	This option specifies the name of the log file generated by the Server. The token %1 may be included to add the value in the “name” configuration option of the [Java-n] section to the file name. The token %2 may be added to include the user ID for the Client in the name. Example: “%1.%2.httpxml.log” will produce a file name similar to: “MyJVM.Smith.httpxml.log”
constantsFile	Valid file name or path and file name to constants file. Default: httpxml_sd.ini	This option names the constants file used for this system connection. This file contains constant named values that may be referenced by the HTTP-XML components of the application.
debug	Boolean value of “true” or “false.” Default: <i>see description.</i>	This option specifies whether debugging messages are generated by the Server in relation to the HTTP-XML System Connection for the [HTTPXML-n] section. This option is set by default to true for development servers and false for production servers.
httpConnectTime-out	Positive integer treated as milliseconds	This option specifies the time out period in milliseconds for this system connection. If this time out value is exceeded the connection will be closed. This timer resets when data is exchanged between Server and the back end system during Client synchronization.
enableAuthentication	Boolean value of “true” or “false.” Default: true	This value specifies whether or not users are authenticated against the back end system for this system connection. At least one system connection within the application must perform user authentication.
xmlValidateOn-Parse	Boolean value of “true” or “false.” Default: false	This option specifies whether or not to validate the XML against the known schema when returned after a request. If false, this validation is not made.

Configuration Settings	Acceptable Values	Description
saveRequest	Boolean value of "true" or "false." Default: <i>see description</i>	This option specifies whether or not to write the request made to the back end system to the log file for the system connection. The defaults for this option are true for a development server and false for a production server.
saveResult	Boolean value of "true" or "false." Default: <i>see description</i>	This options specifies whether or not to write the resulting messaging of a request to the log file for the system connection. The defaults for this option are true for a development server and false for a production server.

Agentry Server: [File-*n*] Section

The following table lists the available configuration options, acceptable values, and descriptions of the settings in the File-*n* set of properties found in the SAP Control Center. These settings are also those found in the [File-*n*] section of the Agentry.ini file.

When adding such a section the *n* portion of the section name must match the ID of the corresponding NT File System connection definition in the application. If no such system connection exists, the settings in this section will not affect the behavior of the Server or the application.

Configuration Setting	Acceptable Values	Description
enableAuthentication	Boolean value of "true" or "false." Default: false	This value specifies whether or not users are authenticated against the host system for this system connection. At least one system connection within the application must perform user authentication.
enablePreviousUserAuthentication	Boolean value of "true" or "false." Default: false	This value specifies whether or not previous users are authenticated against the host system for this system connection. This authentication occurs when a user change occurs on the Client. If allowPreviousUserLogin is false, this setting has no affect.
userDomain	Text value of a configured user domain.	This setting can contain the domain under which all users will be logged into the host system

Configuration Setting	Acceptable Values	Description
tempPath	Valid full path.	This setting specifies the location where temporary files produced by the Server for file system processing are stored. This is normally the batch files defined within the application. If this option is null or not listed in the section the configured Windows temporary folder is used.
allowPreviousUserLogin	Boolean value of “true” or “false.” Default: true	This setting controls whether or not previous users are logged into the file system when a user change occurs on the Client. If not logged in, no file system connection processing defined in the application is performed for the previous user.
name	Text value.	This setting can contain any text value used to identify the system connection in log files and other areas. This should be set to a unique value especially when working with multiple system connections.
debug	Boolean value of “true” or “false.” Default: <i>see description.</i>	This option controls whether or not log file information is generated by the Server for the system connection. The defaults for this option are false for development servers and true for production servers.

Starting, Stopping, or Restarting Agency Server Instances

Start an Agency Server instance to make an enabled application available to end users. Stop an Agency Server instance to perform maintenance or to apply changes to server configuration settings. The Agency Server instance generates log files when it is stopped or started. You can view the logs files in `SMP_HOME\Servers\UnwiredServer\logs\<agencyApplicationID>`.

You might see any of these statuses for an Agency Server instance:

- STARTING
- RUNNING
- RUNNING_WITHOUT_METADATA (the application file has not been deployed)
- RUNNING_NEED_RESTART (to enable a configuration change to take effect)
- STOPPING
- STOPPED
- ERROR

1. In the navigation pane of SAP Control Center, expand the **Applications** node and select the application.
2. In the administration pane, click the **General** tab.
3. Click **Start** , **Stop** , or **Restart** as required.
 - **Start** is available if the application is enabled, and the Agency Server instance status is STOPPING or STOPPED.
 - **Restart** is available if the application is enabled, and the Agency Server instance is STARTING, RUNNING, RUNNING_WITHOUT_METADATA, RUNNING_NEEDS_RESTART, or ERROR.
 - **Stop** is available if the Agency Server instance is STARTING, RUNNING, RUNNING_WITHOUT_METADATA, RUNNING_NEEDS_RESTART, or ERROR.

Startup.log File

The startup.log file contains messages generated by the Agency Server during startup.

The startup.log for the most recent Server execution is found in the folder for the Agency application at the location C:\SAP\MobilePlatform\Servers\UnwiredServer\logs. Previous startup logs are located in backup folders named startuplogs-yyyyymmdd-hh24mmss. The folder is created at startup. The date and time stamp for this folder name is the current startup time. Its contents are the log files from the previous execution of the Server. Information in the startup log includes messages related to loading system connections, front end (Client-Server communications) components, connecting to back end systems, and other tasks performed by the Server during startup.

The Agency Server will always create the startup.log when it is executed. It cannot be disabled.

Shutdown.log File

The shutdown.log file contains messages generated by the Agency Server during shutdown.

The shutdown log for the most recent Agency Server shutdown is found in the folder for the Agency application at the location C:\SAP\MobilePlatform\Servers\UnwiredServer\logs. Previous shutdown logs are located in backup folders named shutdownlogs-yyyyymmdd-hh24mmss. The folder is created at shutdown. The date and time stamp for this folder name is the current shutdown time. Its contents are the log files from the previous execution and shutdown of the Server. Information in the shutdown log includes messages related to disconnecting from system connections, front end (Client-Server communications) components, and other tasks performed by the Server during shutdown.

The Agency Server will always generate the shutdown.log when shutting down. It cannot be disabled.

Native Notification Properties

(Not applicable to Agency applications) The SAP Mobile Server HTTP listener provides a notification interface through which delivery of both native notifications and payload push notifications is configured for a connection. Configure each set of properties accordingly.

Use native notifications to allow third-party applications or an EIS to deliver notifications/payload directly through the SAP Mobile Platform HTTP notification channel to the device. For example, BlackBerry (BES), Apple (APNS), or Android (GCM). Do not confuse native notifications with SAP Mobile Server initiated push notifications.

See also

- *Creating Object API Application Definitions* on page 254
- *Creating Hybrid App Definitions* on page 258
- *Creating REST API Application Definitions* on page 260
- *Creating OData Application Definitions* on page 264
- *Creating Agency Application Definitions* on page 267
- *Application Customization Resource Bundles* on page 296
- *Configuring Native Notifications* on page 255

APNS Native Notification Properties

Set Apple Push Notification Service (APNS) native notification properties to allow delivery of notifications through an SAP Mobile Server HTTP listener to Apple devices.

Note: When configuring the Apple Push Notification Service native notification properties, change the push notification server, push notification port, feedback server, and feedback port values only when configuring notifications in a development environment. To enable Apple push notifications, the firewall must allow outbound connections to Apple push notification servers on default ports 2195 and 2196.

Property	Description
Server	The push notification server.
Port	Push notification server port.
Feedback server	If a feedback service is enabled, the server to which APNS routes feedback information.
Feedback port	The feedback service port.
Certificate (encoded)	The security certificate used for authentication.
Certificate password	The security certificate password.

BES Native Notification Properties

Set BlackBerry Enterprise Server (BES) native notification properties to allow delivery of notifications through an SAP Mobile Server HTTP listener to BlackBerry devices.

These properties are used to authenticate your SAP Mobile Platform to BES Push Service for addressing push request messages to the push-enabled application.

Property	Description
Server URL	Address in the form <i>http://<DNS or IP address>:<portNumber>/pap</i> to push notifications to the device.
User	(Optional) User accessing the URL.
Password/Confirmed Password	(Optional) User password used to connect to the URL.

GCM Native Notification Properties

Set Google Cloud Messaging (GCM) native notification properties to allow delivery of notifications through an SAP Mobile Server HTTP listener to Android devices.

Property	Description
URL	The URL of the push notification server. This field is read-only, and the default value is <i>https://android.googleapis.com/gcm/send</i> .
API key	An API key that is saved on SAP Mobile Server that grants SAP Mobile Platform authorized access to Google services.

Application Customization Resource Bundles

(Does not apply to Hybrid App or Agentry SDK clients) For supported application types, customization resource bundles enable you to associate deployed client applications with different versions of customization resources.

A customization resource bundle can be either:

- **JAR file** – includes a manifest file of name and version properties
- **ZIP file** – must contain META-INF\MANIFEST.MF file, which includes name and version properties:

```
Customization-Resource-Bundle-Name:<name>
Customization-Resource-Bundle-Version:<version>
```

The customization resource bundle does not contain any information that binds or helps bind to applications; it can be uploaded or exported during the definition of an application with SAP Control Center. A deployed customization resource bundle is read-only.

Implementing a customization resource bundle requires the coordination of various roles:

1. (Application developer) Invokes the SDK API that downloads the customization resource bundle. Use the `onCustomizationBundleDownloadComplete` (in the Application Callback API) and `BeginDownloadCustomizationBundle` (in the `SUPApplication` class) methods to pair the application with the device, and reach the client application. See *Developer Guide: OData SDK* or any of the *Object API Developer Guides*.
For example, for an application called `SAP.Mobile.Application`, you might implement the customization resource bundle invocation as follows:

```

/// <summary>
/// start downloading default resource bundle associated with the
/// application. The resource bundle would be saved into writer stream
/// provided by user.
/// an application only bundle an resource
/// </summary>
/// <param name="writer">a writer stream provided by user
/// </param>
public void BeginDownloadCustomizationBundle(System.IO.Stream
writer) { }

/// <summary>
/// start downloading resource bundle named customizationBundleID.
/// The resource bundle would be saved into writer stream provided by
/// user.
/// </summary>
/// <param name="customizationBundleID">the resource bundle name
/// </param>
/// <param name="writer">a writer stream provided by user
/// </param>
public void BeginDownloadCustomizationBundle(string
customizationBundleID, System.IO.Stream writer) { }
Sybase.Mobile.IApplicationCallback

/// <summary>
/// Invoked when download resource bundle complete.
/// </summary>
/// <param name="customizationBundleID">! the resource bundle
/// name. if null, application default resource bundle is downloaded
/// </param>
void OnCustomizationBundleDownloadComplete(string
customizationBundleID, int errorCode, string errorMessage);

```

2. (Developer) Generates the JAR or ZIP with the MANIFEST.MF, which includes these required properties:
 - Customization-Resource-Bundle-Name
 - Customization-Resource-Bundle-Version
3. (Administrator) Uses SAP Control Center to upload the customization resource bundle to SAP Mobile Server then assign it to an application connection.
4. Once the application activation process completes, the application is directed to the appropriate version of the resource bundle.

See also

- *Creating Object API Application Definitions* on page 254
- *Creating Hybrid App Definitions* on page 258
- *Creating REST API Application Definitions* on page 260
- *Creating OData Application Definitions* on page 264
- *Creating Agentry Application Definitions* on page 267
- *Native Notification Properties* on page 295

Application Customization Resource Bundle Recommendations

There are a variety of recommendations for working with customization resource bundles.

- You cannot use customization resource bundles for Hybrid App and Agentry SDK clients.
- The expected format of the customization resource bundle is a JAR archive that contains `MANIFEST.MF`.
 - The manifest file must include these properties:
 - `Customization-Resource-Bundle-Name`
 - `Customization-Resource-Bundle-Version`
 - Property values cannot include a colon (":"), a question mark ("?"), a number sign ("#"), or a percent sign ("%").
 - File size should not exceed 5MB. File size is not enforced, but the larger the file, the slower the performance, and can be subject to device platform hardware capabilities.

See *Managing Application Customization Resource Bundles* in *Developer Guide: SAP Mobile Server Runtime* for information about the administration API that allows programmatic access to this functionality.

- You can assign the same customization resource bundle to different application connections, and it is treated independently for each application that is paired with the connection template that identifies the bundle and version. The primary key is: application ID, customization resource bundle name, and version.
- Each customization resource bundle:
 - Belongs to one and only one application. If you delete an application, all associated customization resource bundles are deleted as well. This implies that the actual binary is stored twice when assigned to two application IDs.
 - Is applicable only to:
 - The application to which it belongs.
 - The application connections that have the same application ID.
 - The application connection templates that have the same application ID.
 - Takes effect only when it is assigned to one or more application connections.
 - Is by default, not assigned to either application connections or application connection templates.
 - Must be assigned explicitly by configuring an application connection or application connection template to use a customization resource bundle.

Note: The application connection assignment configuration overrides that of the application connection template.

- Can be exported to the same JAR file being uploaded to SAP Mobile Server, meaning the format does not change.
- Can upload more than one customization resource bundles, as long as the name and version combination is unique.
- Can assign only one primary customization resource bundle in an application definition. However, any uploaded customization resource bundle is accessible to any application connection.
- You can delete a customization resource bundle only if it is not assigned to any application connection and application connection template with the same application.

Uploading Application Customization Resource Bundles

(Does not apply to Hybrid App or Agentry SDK clients) Before you can assign application customization resource bundles, you must upload them to SAP Mobile Server with SAP Control Center.

Only platform administrators can upload bundles to SAP Mobile Server.

1. In the navigation pane of SAP Control Center, expand the **Applications** node and select the application.
2. In the right administration pane, select the **Customization Resource Bundles** tab.
3. (Optional) Click **Refresh** to update the list of deployed customization resource bundles for this application.
4. (Optional) Add another customization resource bundle to the list.
 - a) Click **Add**.
 - b) In the file dialog, navigate to and select the customization resource bundle JAR file, and click **Open**. The name and version of the newly deployed customization resource bundle is added to the list.
 - c) (Optional) In the Confirm dialog, select one or more check boxes to assign the newly uploaded bundle to application connections or application connection templates with the same application ID. If no check boxes are selected, there is no automatic assignment.

Note: You can make these assignments at a later time.

- d) Click **OK**.
5. (Optional) Add another customization resource bundle to the application.
6. Click **OK**.

Assigning an Application Customization Resource Bundle to a Connection and Template

Assign a customization resource bundle to an individual application connection and application connection template.

1. In the navigation pane of SAP Control Center, expand the **Applications** node and select the application.
2. Select the **Application Connections** or **Application Connection Templates** tab.
3. Select an application connection (user) or application connection template, depending on the tab selected, and click **Properties**.
4. Select **Application Settings**.
5. Select a value from Customization Resource Bundles. These are customization resource bundles that are deployed to the selected application identifier. To unselect a customization resource bundle, select an empty item.
6. Click **OK**.

Assigning an Application Customization Resource Bundle to All Connections Associated with an Application

Assign a customization resource bundle to all application connections and application connection templates associated with a specific application identifier.

1. In the navigation pane of SAP Control Center, expand the **Applications** node and select the application.
2. In the right administration pane, select the **Customization Resource Bundles** tab.
3. (Optional) Click **Refresh** to update the list of deployed customization resource bundles for this application.
4. Select one customization resource bundle to enable the **Assign** and **Unassign** buttons.
5. Click **Assign** to launch the dialog. The list of assignable application connections and application connection templates appears in the respective tabs.
Click **OK** to confirm the assignment for the customization resource bundle, then **Yes**.

Unassigning Application Customization Resource Bundles

Unassign a customization resource bundle from an application.

1. In the navigation pane of SAP Control Center, expand the **Applications** node and select the application.
2. In the right administration pane, select the **Customization Resource Bundles** tab.
3. (Optional) Click **Refresh** to update the list of deployed customization resource bundles for this application.

4. Select the customization resource bundle that you want to unassign.
5. Click **Unassign** to launch the dialog. The list of unassignable application connections and application connection templates appears in the respective tabs.

Click **OK**.

Click **Yes** to confirm the unassignment for the customization resource bundle.

Managing Deployed Application Customization Resource Bundles

Use SAP Control Center to view deployed customization resource bundles, and to export and delete customization resource bundles.

Viewing Deployed Application Customization Resource Bundles

(Does not apply to Hybrid App or Agentry SDK clients) View the customization resource bundles deployed to an application.

1. In the navigation pane of SAP Control Center, expand the **Applications** node and select the application.
2. In the right administration pane, select the **Customization Resource Bundles** tab.
3. You can view the list of deployed customization resource bundles (if any) for the selected application.
4. (Optional) Click **Refresh** to update the list.

Exporting Application Customization Resource Bundles

Export a customization resource bundle from an application.

1. In the navigation pane of SAP Control Center, expand the **Applications** node and select the application.
2. In the right administration pane, select the **Customization Resource Bundles** tab.
3. (Optional) Click **Refresh** to update the list of deployed customization resource bundles for this application.
4. Select a single customization resource bundle in the list and click **Export**.
5. Click **Next**, and then **Finish**.
6. In the file dialog, enter a location and name for the customization resource bundle JAR file, then click **Save**.

Note: The default file name is `CustomizationResourceBundle.jar`. Enter a different name to more specifically identify the resource bundle.

7. (Optional) Export another customization resource bundle JAR file for the application.

Deleting an Application Customization Resource Bundle

Delete a customization resource bundle from an application. You cannot delete a customization resource bundle if it is assigned to an application connection or application connection template; you must unassign it first.

Note: You may have multiple versions of an application in use at once, so it is acceptable to have multiple customization resource bundle versions in the repository. However it is good practice to delete customization resource bundles once you know they are not used.

1. In the navigation pane of SAP Control Center, expand the **Applications** node and select the application.
2. In the right administration pane, select the **Customization Resource Bundles** tab.
3. (Optional) Click **Refresh** to update the list of deployed customization resource bundles for this application.
4. Select a single customization resource bundle in the list and click **Delete**.
5. Click **Yes** to confirm deletion.
6. If the customization resource bundle JAR file is not assigned, it is deleted from the file repository. Otherwise, you must unassign the customization resource bundle first.

Maintaining Activated Applications

Activated applications can be managed and have usage monitored.

Viewing Assigned Connections

(Not applicable to Agentry applications) View the properties of the connections assigned to an application.

1. In the navigation pane of SAP Control Center, select the **Applications** node.
2. In the right administration pane, select the **Applications** tab.
3. Select the application from the list.
4. Click **Application Connections**.
5. Click **Go** to refresh the list that displays the application connections.
6. To filter the list, click **Show filter** and add filters.
7. To define a filter:
 - a) Select the type of information you want to filter on.
 - b) Enter the filter criteria.
 - c) Click +.
 - d) Repeat for additional filters, as required.
 - e) When you have defined all your filters, click **Refresh**.

Different types of filters are combined using a logical AND. If the same type of filter is added multiple times with different criteria, the filters are combined using a logical OR

If you defined the following filters:

- Security Configuration = Security1
- User = User1
- User = User2

The filtered list would return results that matched both Security1 and either User1 or User2.

See also

- *Viewing Assigned Application Users* on page 303
- *Enabling and Disabling Agency Applications* on page 304
- *Viewing Agency Application Active Users* on page 305
- *Viewing Agency Server Instance Information* on page 305
- *Viewing Agency Server Instance Operating System Information* on page 306
- *Rolling Agency Message and Event Log Files* on page 307
- *Configuring Agency Server Instance Logs* on page 307
- *Viewing Correlated Application Details* on page 311
- *Refreshing the Application View* on page 311
- *Deleting Applications* on page 312
- *Searching for Applications* on page 312
- *Monitoring Users of Applications* on page 314

Viewing Assigned Application Users

View the list of the users assigned to an application

1. In the navigation pane of SAP Control Center, select the **Applications** node.
2. In the right administration pane, select the **Applications** tab.
3. Select the application from the list.
4. Click **Application Users**.
5. Click **Go** to refresh the list that displays the application users.
6. To filter the list, click **Show filter** and add filters.
7. To define a filter:
 - a) Select the type of information you want to filter on.
 - b) Enter the filter criteria.
 - c) Click +.
 - d) Repeat for additional filters, as required.
 - e) When you have defined all your filters, click **Refresh**.

Different types of filters are combined using a logical AND. If the same type of filter is added multiple times with different criteria, the filters are combined using a logical OR

If you defined the following filters:

- Security Configuration = Security1
- User = User1
- User = User2

The filtered list would return results that matched both Security1 and either User1 or User2.

See also

- *Viewing Assigned Connections* on page 302
- *Enabling and Disabling Agency Applications* on page 304
- *Viewing Agency Application Active Users* on page 305
- *Viewing Agency Server Instance Information* on page 305
- *Viewing Agency Server Instance Operating System Information* on page 306
- *Rolling Agency Message and Event Log Files* on page 307
- *Configuring Agency Server Instance Logs* on page 307
- *Viewing Correlated Application Details* on page 311
- *Refreshing the Application View* on page 311
- *Deleting Applications* on page 312
- *Searching for Applications* on page 312
- *Monitoring Users of Applications* on page 314

Enabling and Disabling Agency Applications

To start the Agency Server instance for an application, the application must be enabled. Agency applications can be enabled automatically when defining the application or later. If you disable an application when the Agency Server instance is running, the application is no longer available to end users.

1. In the navigation pane of SAP Control Center, expand the **Applications** node and select the application.
2. In the administration pane, click the **General** tab.
3. Click **Enable** or **Disable** as required.
4. Click **Yes**.

See also

- *Viewing Assigned Connections* on page 302
- *Viewing Assigned Application Users* on page 303
- *Viewing Agency Application Active Users* on page 305
- *Viewing Agency Server Instance Information* on page 305
- *Viewing Agency Server Instance Operating System Information* on page 306
- *Rolling Agency Message and Event Log Files* on page 307
- *Configuring Agency Server Instance Logs* on page 307

- *Viewing Correlated Application Details* on page 311
- *Refreshing the Application View* on page 311
- *Deleting Applications* on page 312
- *Searching for Applications* on page 312
- *Monitoring Users of Applications* on page 314

Viewing Agentry Application Active Users

Once an Agentry application is deployed, you can view its active users. Active users are authenticated through the Agentry Server instance for the application. Agentry application login requests are forwarded by SAP Mobile Platform to the Agentry Server instance for authentication.

1. In the navigation pane of SAP Control Center, expand the **Applications** node and select the application.
2. In the administration pane, click the **Active Users** tab.

See also

- *Viewing Assigned Connections* on page 302
- *Viewing Assigned Application Users* on page 303
- *Enabling and Disabling Agentry Applications* on page 304
- *Viewing Agentry Server Instance Information* on page 305
- *Viewing Agentry Server Instance Operating System Information* on page 306
- *Rolling Agentry Message and Event Log Files* on page 307
- *Configuring Agentry Server Instance Logs* on page 307
- *Viewing Correlated Application Details* on page 311
- *Refreshing the Application View* on page 311
- *Deleting Applications* on page 312
- *Searching for Applications* on page 312
- *Monitoring Users of Applications* on page 314

Viewing Agentry Server Instance Information

Each Agentry application has its own Agentry Server instance that runs on the SAP Mobile Server node. Once an Agentry application is deployed, you can review system information for the Agentry Server instance.

1. In the navigation pane of SAP Control Center, expand the **Applications** node and select the application.
2. In the administration pane, click the **General** tab.
3. Click the button in the **Info** column.

See also

- *Viewing Assigned Connections* on page 302
- *Viewing Assigned Application Users* on page 303
- *Enabling and Disabling Agency Applications* on page 304
- *Viewing Agency Application Active Users* on page 305
- *Viewing Agency Server Instance Operating System Information* on page 306
- *Rolling Agency Message and Event Log Files* on page 307
- *Configuring Agency Server Instance Logs* on page 307
- *Viewing Correlated Application Details* on page 311
- *Refreshing the Application View* on page 311
- *Deleting Applications* on page 312
- *Searching for Applications* on page 312
- *Monitoring Users of Applications* on page 314

Viewing Agency Server Instance Operating System Information

Each Agency application has its own Agency Server instance that runs on the SAP Mobile Server node. Once an Agency application is deployed, you can review operating system information for the Agency Server instance.

1. In the navigation pane of SAP Control Center, expand the **Servers** node and select the server.
2. In the administration pane, click the **Properties** tab.
The Agency Server instance operating system information appears at the end of the list of properties.

See also

- *Viewing Assigned Connections* on page 302
- *Viewing Assigned Application Users* on page 303
- *Enabling and Disabling Agency Applications* on page 304
- *Viewing Agency Application Active Users* on page 305
- *Viewing Agency Server Instance Information* on page 305
- *Rolling Agency Message and Event Log Files* on page 307
- *Configuring Agency Server Instance Logs* on page 307
- *Viewing Correlated Application Details* on page 311
- *Refreshing the Application View* on page 311
- *Deleting Applications* on page 312
- *Searching for Applications* on page 312
- *Monitoring Users of Applications* on page 314

Rolling Agentry Message and Event Log Files

Each Agentry application has its own Agentry Server instance that runs on the SAP Mobile Server node. Each Agentry Server instance generates message and event logs for the application, which can be viewed through the Domains folder in SAP Control Center. Agentry message and event log files are automatically rolled based on the configuration of the Agentry application server instance. You can also manually roll the log files.

When you roll an Agentry application log file, the SAP Mobile Server renames the `messages.log` and `events.log` based on the current date and time, and copies them to the backup folders `mlog` and `elog`, respectively. The log files are located in `SMP_HOME\Servers\UnwiredServer\logs\<agentryApplicationID>`.

1. In the navigation pane of SAP Control Center, expand the **Applications** node and select the application.
2. In the administration pane, click the **Logs** tab.
3. Click **Roll log files**.
4. Click **Yes** to confirm.

See also

- *Viewing Assigned Connections* on page 302
- *Viewing Assigned Application Users* on page 303
- *Enabling and Disabling Agentry Applications* on page 304
- *Viewing Agentry Application Active Users* on page 305
- *Viewing Agentry Server Instance Information* on page 305
- *Viewing Agentry Server Instance Operating System Information* on page 306
- *Configuring Agentry Server Instance Logs* on page 307
- *Viewing Correlated Application Details* on page 311
- *Refreshing the Application View* on page 311
- *Deleting Applications* on page 312
- *Searching for Applications* on page 312
- *Monitoring Users of Applications* on page 314

Configuring Agentry Server Instance Logs

Each Agentry application has its own Agentry Server instance that runs on the SAP Mobile Server node. The Agentry Server instance can generate several log messages that are written to one or more log files. You specify the level of verbosity for the log messages and whether the messages should be written to a separate log file. You can view the log files for an Agentry Server instance in `SMP_HOME\Servers\UnwiredServer\logs\<agentryApplicationID>`.

1. In the navigation pane of SAP Control Center, expand the **Applications** node and select the application.
2. In the administration pane, click the **Logs** tab.
3. Select the log for which to configure settings.
4. Click the **Level** for the selected log.
5. Select a verbosity level for the generated log messages.
6. To record the messages to a separate file, in addition to the thread files, click the **Separate File** for the selected log record and select **Yes**.
7. Click **Save**.

See also

- *Viewing Assigned Connections* on page 302
- *Viewing Assigned Application Users* on page 303
- *Enabling and Disabling Agency Applications* on page 304
- *Viewing Agency Application Active Users* on page 305
- *Viewing Agency Server Instance Information* on page 305
- *Viewing Agency Server Instance Operating System Information* on page 306
- *Rolling Agency Message and Event Log Files* on page 307
- *Viewing Correlated Application Details* on page 311
- *Refreshing the Application View* on page 311
- *Deleting Applications* on page 312
- *Searching for Applications* on page 312
- *Monitoring Users of Applications* on page 314

Agency Server Logs Overview

You can use the log messages generated by the Agency Server to diagnose development or production issues, determine processing times for numerous aspects of synchronization, and to obtain various other pieces of information.

The Agency Server can generate several log messages that are written to one or more log files. There are different categories of log file messages that pertain to a different aspect of the application:

- Client-Server communications
- System connection communications
- User-specific information for both Client-Server and system connection synchronization
- Definition loading and processing
- Code file (SQL, Java, batch files, etc.) processing, including post-SDML expansion and results, JVM messages, and so on
- Agency Server system log files (these log files are always generated by the server and have no configurable behaviors)

Agentry Server System Log Files

Following are the system log files for the Agentry Server:

- `messages.log*`
- `events.log*`
- `startup.log`
- `shutdown.log`

** - These log files can be viewed using the SAP Control Center.*

These log files are always generated and cannot be disabled. Their contents do not have different verbosity levels.

Log Message Contents and Categories

Log files are enabled or disabled, configured, and managed using the SAP Control Center. The options available are to enable or disable the various log files, to specify whether log files are generated based on a processing thread or more narrow areas of functionality, and to set the verbosity level of the log message contents.

The log messages generated by the Agentry Server are categorized as follows:

- **Server:** Messages generated as a result of Agentry Server processing. The contents are an inclusive set of the other categories, with the messages including all those related to any processing performed by the Agentry Server.
- **User:** Messages generated that are specific to a single user. The contents are an inclusive set of the other categories, with the messages including all those related to any user-specific processing performed by the Agentry Server.
- **System Connection:** Messages generated resulting from processing performed by the Agentry Server for back end data synchronization for a specific system connection. The contents are messages that result from processing the various synchronization definitions within the application. Each system connection within an application has its own set of log messages that you can individually enable and configure.
- **Client Communications:** Messages generated resulting from Client-Server communications. These messages include listings of involved IP addresses and port numbers, opening of sockets, transfer of data in both directions, and other messages sent between the Agentry Client and Agentry Server.

The contents of all log messages, regardless of category, include both error messages and informational or status messages. Each message includes a date and time stamp precise to the millisecond. The verbosity of the messages generated is configured when the category of log messages are enabled.

Log Messages Versus Log Files

Log messages generated by the Agentry Server are stored in *log files*. When enabling and configuring the logging behavior of the Agentry Server, the contents of the log messages are configured. In addition, the files into which those messages are written is also configured. By

default, when a log message category is enabled, those messages are written to a log file for a processing thread. Each Agentry Server processing thread will have its own log file containing all messages generated by the Agentry Server for processing within that thread, and for the category of log messages that have been enabled.

As an optional additional method for storing log messages, each category of log messages can be written to a category-specific log file. When this is enabled, all log messages generated by the Agentry Server for a specific category are written to the same log file, regardless of processing thread. The messages within the category-specific log files are grouped by the processing thread. Therefore, a category log file contains log messages resulting from processing related to that category.

Following are the individual log files that can be generated for each category, in addition to the thread log files:

- **Server category:** `Server-ServerName.log` where *ServerName* is the configured name for the Agentry Server in the `Agentry - systemConnection` configuration option. The category is `Agentry` by default.
- **User category:** `User-UserID.log` where *UserID* is the user's Agentry Client login ID.
- **System Connection category:** `BackEnd-SystemConnection.log` where *SystemConnection* is the name of the system connection section (i.e., `SQL-1`, `Java-2`).
- **Client Communications category:** `FrontEnd-FrontEndType.log` and `ANGEL Connection-www.xxx.yyy.zzz.log` where *FrontEndType* is the connect type (i.e., `ANGEL Front End`, `Web Front End`). The second file is specific to the ANGEL Connection type. One file is created for each Agentry Client where `www.xxx.yyy.zzz` is the Client's IP address as seen by the Agentry Server.

Configuring the log files to generate separately based on the log message category will have a negative impact on Agentry Server performance, as multiple processing threads are logging messages to the same physical log file. Only a single message is written at a time to such a log file. Therefore, each thread must wait until the log file is available before it can write its log message to the log file. It is recommended to configure log files in this manner only for Agentry Development Servers with any regularity. It can be done in a production environment, but only when necessary and for limited durations.

All log files are written to the applicable Agentry application folder in `C:\SAP\MobilePlatform\Servers\UnwiredServer\logs`. Log files are rolled periodically by the Agentry Server, during Agentry Server startup or restart, and on command. The files are rolled to the sub-directory `Logs-Rolled`.

Viewing Correlated Application Details

Select an application, then view correlated application details in several categories, including application users and application connections.

1. In the left navigation pane, click the **Applications** node.
2. In the right administration pane, click the **Applications** tab.
You can view the list of registered applications.
3. Select one application from the list.
4. Select one of the buttons to view related application users or application connections.
 - **Application Users** – view correlated application users.
 - **Application Connections** (Not applicable to Agentry applications) – view correlated application connections.
5. In Review Assignment, check the information.
6. Click **OK**.

See also

- *Viewing Assigned Connections* on page 302
- *Viewing Assigned Application Users* on page 303
- *Enabling and Disabling Agentry Applications* on page 304
- *Viewing Agentry Application Active Users* on page 305
- *Viewing Agentry Server Instance Information* on page 305
- *Viewing Agentry Server Instance Operating System Information* on page 306
- *Rolling Agentry Message and Event Log Files* on page 307
- *Configuring Agentry Server Instance Logs* on page 307
- *Refreshing the Application View* on page 311
- *Deleting Applications* on page 312
- *Searching for Applications* on page 312
- *Monitoring Users of Applications* on page 314

Refreshing the Application View

Refresh the list of all available applications registered through the SAP Control Center.

1. In the navigation pane of SAP Control Center, select the **Applications** node.
2. In the right administration pane, select the **Applications** tab.
3. To view the list of registered applications, click **Refresh**.

See also

- *Viewing Assigned Connections* on page 302
- *Viewing Assigned Application Users* on page 303

- *Enabling and Disabling Agency Applications* on page 304
- *Viewing Agency Application Active Users* on page 305
- *Viewing Agency Server Instance Information* on page 305
- *Viewing Agency Server Instance Operating System Information* on page 306
- *Rolling Agency Message and Event Log Files* on page 307
- *Configuring Agency Server Instance Logs* on page 307
- *Viewing Correlated Application Details* on page 311
- *Deleting Applications* on page 312
- *Searching for Applications* on page 312
- *Monitoring Users of Applications* on page 314

Deleting Applications

Delete an application to remove all the registered users, connections, and subscriptions associated with those connections. Delete an application to remove all associated runtime artifacts on the server. Deleting applications removes application definitions, application users and connections associated with the application, and package-level subscriptions of the application connections. Delete applications with care to avoid adversely impacting users.

1. In the left navigation pane of SAP Control Center, click the **Applications** node and select the **Applications** tab in the right administration pane.
2. Select the application and click **Delete**.

See also

- *Viewing Assigned Connections* on page 302
- *Viewing Assigned Application Users* on page 303
- *Enabling and Disabling Agency Applications* on page 304
- *Viewing Agency Application Active Users* on page 305
- *Viewing Agency Server Instance Information* on page 305
- *Viewing Agency Server Instance Operating System Information* on page 306
- *Rolling Agency Message and Event Log Files* on page 307
- *Configuring Agency Server Instance Logs* on page 307
- *Viewing Correlated Application Details* on page 311
- *Refreshing the Application View* on page 311
- *Searching for Applications* on page 312
- *Monitoring Users of Applications* on page 314

Searching for Applications

Search for registered applications from the default view, or perform an advanced search. The advanced search enables you to search through applications, users, application connections,

packages, and subscriptions, filtering out results at each level until you obtain very specific results.

See also

- *Viewing Assigned Connections* on page 302
- *Viewing Assigned Application Users* on page 303
- *Enabling and Disabling Agentry Applications* on page 304
- *Viewing Agentry Application Active Users* on page 305
- *Viewing Agentry Server Instance Information* on page 305
- *Viewing Agentry Server Instance Operating System Information* on page 306
- *Rolling Agentry Message and Event Log Files* on page 307
- *Configuring Agentry Server Instance Logs* on page 307
- *Viewing Correlated Application Details* on page 311
- *Refreshing the Application View* on page 311
- *Deleting Applications* on page 312
- *Monitoring Users of Applications* on page 314

Searching from the Default View

Search for applications that are registered in the SAP Control Center.

1. In the navigation pane of SAP Control Center, select the **Applications** node.
2. In the right administrations pane, select the **Applications** tab.
3. To set the search criteria, select the criteria from the **Search** drop-down list.
4. Add a search string.
5. Click **Go**.
All the applications that match the search criteria provided are populated in the table.

See also

- *Performing an Advanced Search* on page 313

Performing an Advanced Search

search through applications, users, application connections, packages, and subscriptions, filtering out results at each level until you obtain very specific results.

1. In the navigation pane of SAP Control Center, select the **Applications** node.
2. In the right administrations pane, select the **Applications** node.
3. Select **Advanced Search**.
4. In Advanced Search, enter an application ID, then click **Go**, or click **Go** to display a list of applications.

5. Select an application, and click **Search**.
The **Users**, **Application Connections**, **Packages**, and **Subscriptions** tabs remain.
6. Select one of the remaining tabs, select the search criteria, and click **Search**.
7. Continue this process until you have the information you seek. Each time you search in a one of the categories, the tab is removed. In this way you refine your search.

See also

- *Searching from the Default View* on page 313

Monitoring Users of Applications

Track and modify application usage by user, and maintain application users as required.

See also

- *Viewing Assigned Connections* on page 302
- *Viewing Assigned Application Users* on page 303
- *Enabling and Disabling Agency Applications* on page 304
- *Viewing Agency Application Active Users* on page 305
- *Viewing Agency Server Instance Information* on page 305
- *Viewing Agency Server Instance Operating System Information* on page 306
- *Rolling Agency Message and Event Log Files* on page 307
- *Configuring Agency Server Instance Logs* on page 307
- *Viewing Correlated Application Details* on page 311
- *Refreshing the Application View* on page 311
- *Deleting Applications* on page 312
- *Searching for Applications* on page 312

Deleting Application Users

Delete a user to remove the entry as well as personalization data from the cache database.

1. In the left navigation pane of SAP Control Center, click the **Applications** node and select the **Application Users** tab in the right administration pane.
2. Select the user and click **Delete**.

The user entry and data are removed.

Disconnecting Agency Application Active Users

Disconnect Agency application active users when you no longer want them to have access to the application.

1. In the navigation pane of SAP Control Center, expand the **Applications** node and select the application.
2. In the administration pane, click the **Active Users** tab.
3. Select the user you want to disconnect and click **Disconnect**.
4. Click **Refresh**.

The user no longer appears in the list, and can no longer access the application.

Checking Application User Assignments

Check which applications are used by registered users.

1. In the left navigation pane of SAP Control Center, click the **Applications** node and select the **Application Users** tab in the right administration pane.
2. Select an application user and click **Applications**.

All applications used by the user are listed in the dialog.

Searching for Application Users

Search for application users according to the criteria you specify.

1. In the left navigation pane of SAP Control Center, select the **Applications** node.
2. In the right administration pane, select the **Application Users** tab.
3. Choose the type of information that you want to search on from the **Search** drop-down list, then specify the search string .
4. Click **Go**.

The user information is populated according to the criteria you have specified.

Refreshing the Application Users View

Refresh the application user list to display current information about registered users.

1. In the left navigation pane, select the **Applications** folder.
2. In the right administration pane, select the **Application Users** tab.
3. Click **Refresh** to view the current information of all users.

Transporting Applications Between Environments Using Export and Import

Transport applications between different server environments by exporting them from one server, before importing them into another one. This sequence of events is typically performed by administrators when transporting applications from a development local server environment to a test network environment, or from a test or pilot environment to a production cluster environment.

Exporting Applications

Export applications to create a deployment archive that can be used to transport applications between SAP Mobile Servers.

Prerequisites

Before beginning, review import requirements and best practices.

Task

1. In the left navigation pane of SAP Control Center, select **Applications**.
2. In the right navigation pane, click the **Applications** tab.
3. Select the box adjacent to the application and click **Export**.
4. Click **Next**.
5. Click **Finish**.
6. Select the file system target for the exported contents and click **Save**.

Note: Ensure that you do not hide the file type extension when you name the export archive; otherwise, the *.zip extension becomes invisible, which adversely affects the outcome of the export process.

A status message indicates the success or failure of the export transaction. If the transaction succeeds, a ZIP file is created in the location you specified. You can then import this file on another SAP Mobile Server.

Next

Deliver the file to the appropriate person, or deploy or transport the exported application to the appropriate server.

Importing Applications

Import applications after they have been exported from another SAP Mobile Server.

Prerequisites

Review import requirements and best practices before beginning.

Task

Note: Only platform administrators can import applications at the cluster level.

1. In the navigation pane of SAP Control Center, select the **Applications** node.
2. In the right administration pane, click the **Applications** tab and click **Import**.
3. Click **Browse** to navigate to the file.

4. Click **Import**.
5. After the import is complete, check the import details and click **OK**.

Next

If the application connection template does not exist on the target server, it is imported. You may need to change the connection and proxy settings to match the target server configuration. If the application connection template already exists on the target server, connection and proxy settings are not imported; the target server template settings are used for connection and proxy settings.

Import Requirements and Best Practices

Import is typically used to move a package or application from a development environment to a test environment, and after testing to a production environment.

MBO Package Imports

- **Domain requirements** – all server connections and security configurations referenced by the MBO package must exist in the target domain.
- **Versioning recommendations** – if a developer has updated the package version number:
 1. (Required) Verify that this new package version is added to the application.
 2. (Recommended) Whenever possible, use the update instead of import. Otherwise, delete the existing package first, to remove all runtime data for the package including cached data, registered subscriptions, subscription templates, client log, MBO and operation histories, and registered package users. Delete these items only after serious consideration.

Application Imports

Make sure the target system has resources that match those referenced in the export archive file:

- Domains, security configurations, logical roles assigned to application connection templates, proxy endpoint connections (used by ODP applications)
- If MBO packages are included, the domains, security configurations and connections referenced by the MBO package archives

Hybrid App Imports

If MBO packages are referenced in the export archive file, make sure that the MBO packages have already been deployed to the target system.

Note: If the Hybrid App has matching rules, all matching rule search expressions are imported as regular expression types. Other expression types such as `Begins with` or `Equals` are imported as `Regular expression`.

Application Connections

(Not applicable to Agentry applications) Application connections define the manner in which the application connects to SAP Mobile Server and interacts with the runtime services the application requires. Connections are defined by the properties that are configured for it, frequently saved as a reusable template.

The connection registration process is either manual or automatic. Without successful registration, a connection is not activated, and an application cannot consume the data, the customizations, or the packages it requires.

SAP recommends that you use templates to store and reuse connections. At minimum, you should create one template per security configuration.

See also

- *EIS Data Source Connection Properties Reference* on page 170
- *Creating Connections and Connection Templates* on page 168

Registering or Reregistering Application Connections

Registering an application connection groups the user, device, and application to create a unique connection in SAP Control Center, so the registered connection activity can be monitored. Use SAP Control Center to manually register an application connection. You can also reregister an application connection when the association between the user, device and application breaks or requires a different pairing.

For more information on registering and reregistering application connections, see *How Connections Are Registered* in *Mobile Application Life Cycle*.

1. In the left navigation pane, click the **Applications** node.
2. In the right administration pane, click the **Application Connections** tab.
3. Choose an action:
 - Click **Register** to register a new application connection. Using the Activation Code, this application is then paired with a user and a device.
 - Click **Reregister** to associate the application with a new device and user pairing. For example, reregister the application connection if someone loses their device. By reregistering the application connection, the user then receives the same applications and workflows as the previous device.

Note: If the client application does not support reregistration, you cannot reregister the application connection. To determine if the client application supports reregistration, review the **Capabilities** properties for the application connection. If the **Application Supports Client Callable Components** property has a value of `False`, reregistration is not supported.

4. In the Register Application Connection or the Reregister Application Connection dialog.

- a) For new device registration only, type the name of the user that will activate and register the device. For reregistrations or clones, the same name is used and cannot be changed.
- b) (Not applicable to reregistration.) Select the name of the template for initial application connection registration. The template you use supplies initial values in the subsequent fields.
 - Default – a default template that you can use as is, or customize.
 - HWC – a default template for Hybrid Web Container. Use as is, or customize. If you use the HWC template, Application ID must be set to HWC.
 - Custom - customized templates are listed.

Note: You cannot change the application connection template for an application connection after registration.

5. Change the default field values for the template you have chosen.

If you are using Relay Server, ensure the correct values are used.

- **Application ID**- the application ID registered for the application. The value differs according to application client type - native application, Hybrid App, or Online Data Proxy client. See *Application ID Overview* for guidelines.

Note: If the template you have chosen supplies the Application ID, then this field is read-only.

- **Security Configuration** (Registration only) - select the security configuration relevant for the application connection.
- **Logical Role**- (not applicable to reregistration) select the logical role that users must belong to in order to access the application.
- **Domain** (Registration only) - select the domain to which the application connection is assigned. A domain is not required for registering application connections for Hybrid Web Container applications.

Note: This value is sent to and used by the device application, and is automatically derived from the application ID you select. Therefore, you must set this value correctly when using a domain with an application ID. If you set a domain, ensure it matches the domain of the packages needed by the application; otherwise, the application generates a `Package not found` error.

- **Activation code length** - the number of characters in the activation code.
 - **Activation expiration**- the number of hours the activation code is valid.
6. (Optional) Select the check box adjacent to **Specify activation code** to enter the code sent to the user in the activation e-mail. This value can contain letter A - Z (uppercase or lowercase), numbers 0 - 9, or a combination of both. Acceptable range: 1 to 10 characters.
7. Click **OK**

The application is registered or reregistered. SAP applications that have connections registered with SAP Mobile Server, can also have licenses counted by SAP License Audit

service. For a list of SAP applications for which licenses are counted, see *SAP Applications Tracked with SAP License Audit* in *System Administration*.

See also

- *Considerations for Application Connection Registration* on page 322
- *Application Connection Properties* on page 329

Application ID and Template Guidelines

Choose an appropriate application ID while registering application connection for use by native MBO, Hybrid App, or Online Data Proxy clients. Using an incorrect application ID results in failure when the client tries to activate itself.

Application Type	Guidelines
Hybrid App	• 2.0.1 or earlier – leave the application ID empty. • 2.1 or later – use preexisting HWC template, or, if you are using your own template, make sure that HWC is set as the application ID in the template. • iOS sample container 2.1 or later – use the template you have created. The application ID used by the iOS sample container should match the application ID specified in registration.
Native MBO application	• Previous to 2.1.2 – leave the application ID empty. This applies to native messaging-based application clients. • 2.1.2 or later – (recommended) use the application connection template that is automatically created for the application. Otherwise, ensure you register the application connection with the correct template by verifying that application ID matches, and that the correct security configuration and domain are selected. Also, if using replication, set other template properties (such as synchronization-related properties in Connection category) as required. For Android native MBO applications, this recommendation applies starting with version 2.1.1.
Online Data Proxy	Register the application connection using the template created for the application. Existing templates can be found in the Applications > Application Connection Template tab.

Assigning and Unassigning Hybrid Apps to Application Connections

Assign Hybrid App packages to application connections to make them available to device users.

1. In the left navigation pane of SAP Control Center, click **Applications**.
2. In the right administration pane, click the **Application Connections** tab.
3. Select the application connection to assign a Hybrid App package to.
4. Click **Hybrid Apps**.
5. Click **Assign**.
6. Select the Hybrid App package or packages that you want to assign to the application connection.
7. Click **OK**.

Note: If the client application does not support Hybrid Apps, you cannot assign a Hybrid App to the application connection. To determine if the client application supports Hybrid Apps, review the **Capabilities** properties for the application connection. If the **Application Supports Hybrid App** property has a value of `False`, Hybrid Apps are not supported.

8. To set a Hybrid App package as the default application for the application connection, select the package and click **Set Default**.

Set a Hybrid App package as the default to run that application on the device as a single-purpose application. Single-purpose applications launch automatically when the user opens the Hybrid Web Container. You can only select one default per application connection.

9. To unassign a Hybrid App package, select the package and click **Unassign**.

Note: If you unassign the Hybrid App package that is set as the default for the application connection, you may want to select a new default package.

10. Click **OK**.

Maintaining Registered Connections

Connections that are registered can be maintained and managed. Perform any connection maintenance tasks as needed.

Searching for Application Connections

Set search criteria to filter connections viewed in the Application Connections tab

1. In the left navigation pane of SAP Control Center, click the **Applications** node.
2. In the right administration pane, click the **Application Connections** tab.
3. To set the search criteria, configure these search elements:

- Choose the connection information column name you want to enter a search value for.
 - Type or choose the value for the column name you selected.
4. To search based on multiple filter criteria, select **Multi filters** and add filters.
 5. To define a filter:
 - a) Select the type of information you want to filter on
 - b) Enter the filter criteria.
 - c) Click +.
 - d) Repeat for additional filters, as required.
 - e) When you have defined all your filters, click **Go**.

You can define only one filter criteria for each type of information. The filters are combined using a logical AND.

Considerations for Application Connection Registration

An application connection can be registered automatically, manually or anonymously.

Information and guidelines:

- Application templates hold default connection properties that can be assigned to an application during the connection registration process. However, these templates are configured differently depending on what type of connection registration you enable for applications. See the recommendations documented in *Creating Application Connection Templates* in the *Mobile Application Life Cycle* guide.

When a client application connects to SAP Mobile Server its application ID is used to look up a matching template. If that template allows automatic registration (the Automatic Registration Enabled property is set to true), then the security configuration in the template is used to authenticate the user and establish an identity against which the connection is registered. If the template also specifies a logical role, then user is authorized using the mapped physical role(s) of the logical role in the security configuration. When there are templates with different logical roles for the same application id and security configuration, the priority of the template determines the order of evaluation of the associated logical roles.

Note: If no templates are detected, the registration request fails. If multiple templates are detected, the client application registers using the template with the highest priority. If there is more than one template with the highest priority, the application registers using one of the templates with the highest priority, selected at random. For details on how user names and security configuration names are processed when an email address is used, see *Considerations for Using E-mail Addresses as User Names* in the *Security* guide.

- Supported device client activation options:

Device Client Type	Automatic Registration	Manual Registration	Anonymous Registration
Hybrid App	X	X	X

Device Client Type	Automatic Registration	Manual Registration	Anonymous Registration
Object API	X	X	
OData	X	X	X
REST API	X		X

See also

- *Registering or Reregistering Application Connections* on page 318

Deleting Application Connections

Delete an application connection to remove a user assignment to an application connection.

1. In the left navigation pane of SAP Control Center, click the **Applications** node and select the **Application Connections** tab in the right administration pane.
2. Select the application connection and click **Delete**.

Editing the Application Connection Properties

Modify or update the properties of an application connections

1. In the left navigation pane, click the **Applications** node.
2. In the right administration pane, click the **Applicaton Connections** tab.
3. Select an application connection from the list.
4. Click **Properties**.
 - a) In the Application Connection Properties dialog, select the category from the left pane.
 - b) Update or modify the property and its value.

Note:

- You cannot update or modify the **Capabilities** properties. These properties are set by the client application.
 - If the client application does not support a password policy, you cannot update or modify the **Password Policy** properties. To determine if the client application supports a password policy, review the **Application Supports Password Policy** property under **Capabilities**. A value of `False` means a password policy is not supported.
-

- c) Click **OK**.
-

Note: When the application end-point for a registered application is modified under the **Proxy** property, you have to manually update the **Address** in the proxy properties of the connection pool.

Cloning Application Connections

Create a duplicate copy of an application connection configuration settings. This feature allows you to retain user information and pair it with a different device in the event that a user gets a new or alternate device.

1. In the left navigation pane, click the **Applications** node.
2. In the right administration pane, click the **Application Connections** tab.
3. Check the box adjacent to the connection you want to clone and click **Clone**.

Note: If the client application does not support cloning, you cannot clone the application connection. To determine if the client application supports cloning, review the **Capabilities** properties for the application connection. If the **Application Supports Client Callable Components** property has a value of `False`, cloning is not supported.

4. Edit the **Activation expiration** configuration setting associated with the application connection. This setting determines the number of hours the activation code is valid.
5. (Optional) Select the check box adjacent to **Specify activation code** to enter the code sent to the user in the activation e-mail. This value can contain letter A - Z (uppercase or lowercase), numbers 0 - 9, or a combination of both. Acceptable range: 1 to 10 characters.
6. Click **OK**

Viewing Package Users for Application Connections

View users of packages associated with a registered application connection. Users can only be viewed following connection registration. Connection registration creates a pairing between the user, the application ID, and the connection.

1. In the left navigation pane, click the **Applications** node.
2. In the right administration pane, click the **Application Connections** tab.
3. Check the box adjacent to the connection you want to review package users for and click **Package Users**.

Note: If the client application does not support package users, you cannot review package users for the application connection. To determine if the client application supports package users, review the **Capabilities** properties for the application connection. If the **Application Supports Client Callable Components** property has a value of `False`, package users are not supported.

4. View package user information. Select and delete users if required.
5. Click **OK**.

Locking and Unlocking Application Connections

Lock or unlock connections to control which users are allowed to synchronize data. Locking an application connection is an effective way to disable a specific user without making changes to the security profile configuration to which he or she belongs. Locking an

application connection blocks delivery of generated data notifications to the replication-based synchronization clients.

1. In the left navigation pane, select the **Applications** node.
2. In the right administration pane, select the **Application Connections** tab.
3. Select the application connection you want to manage, and:
 - If the connection is currently unlocked and you want to disable synchronization, click **Lock**.
 - If the connection is currently locked and you want to enable synchronization, click **Unlock**.
4. In the confirmation dialog, click **OK**.

Tracing Application Connections

Send a request to the device to retrieve log files for an application connection.

1. In the left navigation pane, select the **Applications** node.
2. In the right administration pane, click **Application Connections** tab.
3. Select an application connection, and click **Get Trace**.

Note: If the client application does not support tracing, you cannot trace the application connection. To determine if the client application supports tracing, review the **Capabilities** properties for the application connection. If the **Application Supports Client Callable Components** property has a value of `False`, tracing is not supported.

The application connection status must be "online" to retrieve the logs.

4. Click **OK**.
5. When the application connection is online, you can view the log contents in SAP Control Center by retrieving the server log for the domain that the application connection belongs to. The trace logs will be identified by one of the following values in the Category column of the Server log tab: PerformanceAgent, MOCA, or SQLTrace. Trace logs can also be viewed in the file system. The default location for single node and cluster installations is `SMP_HOME\Servers\UnwiredServer\logs\ClientTrace`.

Viewing Default Hybrid Apps

Set a Hybrid App package as the default application for an application connection to run on the device as a single-purpose application. A single-purpose application launches automatically when the user opens the Hybrid Web Container and is the only Hybrid App available on the device.

1. In the left navigation pane of SAP Control Center, click **Applications**.
2. In the right administration pane, click the **Application Connections** tab.
3. Select the application connection you want to view the default Hybrid App for.

4. Click **Hybrid Apps**.

The Hybrid Apps assigned to the connection are listed. The default application is identified with a check mark in the **Default** column.

5. To change the default, select the current default Hybrid App and click **Unset default**, then select the new default Hybrid App and click **Set default**.

You can only select one default per application connection.

6. Click **OK**.

Setting Anonymous Access for Applications

To provide anonymous access to a manually created or existing application, select the anonymous security configuration and the Anonymous access checkbox.

1. To set anonymous access for an existing application, select the application and create an application connection template with the Anonymous security configuration.
2. For either new or existing ODP applications, you must also manually create a connection for the application where the:
 - connection pool name is the application ID
 - connection pool type is Proxy
 - Allow Anonymous Access property = true

Application Connection Templates

(Not applicable to Agentry applications) An application connection template is a model or pattern used to standardize properties and values for application connections. When application share many of the same connection properties and values save them to a connection template.

The application connection template assigned during application connection registration sets the default connection with common properties and values. You can then override the default template properties and specific values.

You can also assign Hybrid App packages to an application connection template. When you assign Hybrid App packages to an application connection template, all application connections that use the template are automatically assigned the Hybrid App packages.

Commonly used application properties include: application ID, security configuration, and logical role (used for automatic application connection registration for role based access), and domain. Together, the application ID, security configuration, and logical role form a unique key. An application may have multiple templates that use the same security configuration but different logical roles. During the course of automatic registration and application activation, the **Template Priority** property of in template's **Application Settings** defines the order in which the logical roles are evaluated for user authorization.

Use the built-in templates or create new ones as dictated by your deployment environment. You can also use these built-in templates to customize templates beyond those created from development property values:

- **Default** – Registers application connections without an application ID. Use this option for backward compatibility scenarios, or with previous versions of client runtime for native messaging clients.
- **HWC** – Registers application connections for Hybrid Web Container clients only.
- **\$diagtool** – Registers application connections to verify a valid end-to-end configuration.

Creating Application Connection Templates

Create application connection templates by setting appropriate properties and values.

1. In the left navigation pane, click the **Applications** node.
2. In the right administration pane, click the **Application Connection Templates** tab.
3. Click **New**.
4. Enter the **Template name** and **Description** for the application connection template.
5. Select the **Base template** from the drop-down list.
6. You can configure any of the following profiles. See *Application Connection Properties*:
 - Application Settings
 - Android Push Notifications
 - Apple Push Notifications
 - BlackBerry Push Notifications
 - Capabilities
 - Connection
 - Custom Settings
 - Device Advanced
 - Device Info
 - Password Policy
 - Proxy
 - Security Settings
 - User Registration

Note: To change the password policy, the **CapabilitiesPasswordPolicy** property should be set to true.

To get more details on application connection properties, see *Application Connection Properties* in *SAP Control Center for SAP Mobile Platform*.

7. Click **OK**.

See also

- *Application Settings* on page 330

Assigning and Unassigning Hybrid Apps to Application Connection Templates

When you assign Hybrid App packages to an application connection template, all registered application connections are automatically assigned Hybrid App packages.

You can also assign Hybrid App packages directly to an application connection. The set of packages assigned to an application connection will be a combination of packages assigned indirectly through the application connection template and directly through the application connection..

1. In the left navigation pane, select **Applications** .
2. In the right navigation pane, click the **Application Connection Templates** tab.
All the application connection templates are listed.
3. Select a template and click **Hybrid Apps**.

Note: The selected template must have an application ID.

4. Click **Assign**.
5. Select one or more Hybrid App packages that you want to assign to the template and click **OK**.
6. To set a Hybrid App package as the default for the application connection template, select the Hybrid App package and click **Set default**.

Set a Hybrid App package as the default to if you want it to launch automatically when the user opens the Hybrid Web Container. This will be the only Hybrid App available on the device. You can only select one default per application connection template.

Note: If the same application connection has a default Hybrid App package assigned to it directly (not through the application connection template), the direct assignment default takes precedence.

7. To unassign a Hybrid App, select the Hybrid App and click **Unassign**.

Note: If you unassign the default Hybrid App, you may want to select a new default Hybrid App.

Changing Properties of Application Connection Templates

(Does not apply to Agency SDK clients) Change existing property values used in application connection templates.

Note: Changes made to application connection template properties are applied to all application connections that use that template, except where the properties have been overridden for a specific application connection.

1. Do one of the following in the navigation pane of SAP Control Center:
 - Select the **Applications** node.

- Expand the **Applications** node and select an application.
- 2. In the right administration pane, click the **Application Connection Templates** tab.
- 3. Select the application connection template from the list and click **Properties**.
- 4. In the Template dialog, select the category you want to edit and modify the property and value.

Note: Some properties are read-only and cannot be changed by the administrator. Some read-only properties are set by the application developer or the device. Other read-only properties are set by the administrator but cannot be changed after the template has been used to register an application connection.

5. Click **OK**.

Deleting an Application Connection Template

When an application connection template is no longer needed, delete the template.

Understand the risks and outcomes of this action before proceeding:

- If an application was defined using a template targeted for deletion, then deleting a template prevents future connections from registering. Therefore, before you delete a template, always ensure it is not used in any existing application definition.
 - If the application connection template is used by a registered application connection, you cannot delete the template.
1. In the left navigation pane, click the **Applications** node.
 2. In the right administration pane, click the **Application Connection Templates** tab.
 3. Select the application connection template from the list and click **Delete**.
 4. Click **OK** on the confirmation dialog.

Application Connection Properties

(Not applicable to Agentry applications) Set application connection properties through SAP Control Center to create application connections and application connection templates. These settings influence the application connection, including security, datasource connection, user registration, device and application, and so forth.

See also

- *Registering or Reregistering Application Connections* on page 318

Application Settings

Application settings display application information, including the Application Identifier, Domain, and Security Configuration

- **Automatic Registration Enabled** – set to **True** to automatically register the application using a connection template. Be sure you understand the security ramifications of setting this value to true. See *Registering Applications, Devices, and Users* in the *Security* guide.
- **Application Identifier** – the application identifier registered on SAP Control Center.
- **Application Version** – the version number of the registered application.
- **Client SDK Version** – the version number of the SDK for the registered application.
- **Customization Resource Bundles** – the application configuration (customization resource bundles) associated with the application. Values include:

- A single name, such as `Appmc:1.2.1`, indicates a single customization resource bundle.
- Blank means no customization resource bundles are assigned.

Note: Customization resource bundles are not supported for Hybrid App and Agentry SDK clients.

- **Domain** – the domain selected for the connection template.

The domain is not required when automatic registration is enabled.

- **Notification Mode** – the notification mode through which native notifications and payload push notifications to registered devices are delivered:
 - Only native notifications – allows third party applications or EIS to deliver native notifications directly through the HTTP notification channel: BlackBerry (BES), Apple (APNS), or Android (GCM) to the device.
 - Only Payload(Online) Push – allows third party applications or EIS to deliver data payload push notifications to an online device.
 - Payload(Online) Push with Native Notifications – allows both payload and native notifications to be delivered to the device.

Note: Apple and Google do not recommend payload delivery over their systems:

- Data may not be delivered.
- Data is delivered out of sequence.
- If enabled, the actual payloads must be small.

For example, GCM makes no guarantees about delivery or order of messages. While you might use this feature to inform an instant messaging application that the user has new messages, you probably would not use it to pass actual messages.

RIM supports guaranteed delivery, including callbacks to notify SAP Mobile Server that the message was delivered or when it failed. However, this is a different message format. RIM messaging has many limitations including packet size, number of packets for a single

user that BES keeps, number of packets BES keeps for all users, and so on. Therefore, BES should also only be used to send the notification, but not to send payloads.

Refer to the appropriate platform documentation (APNS, BES, or GCM) for additional information.

- **Security Configuration** – the security configuration defined for the connection template.

The security configuration of the application connection template is used to authenticate users when automatic registration is enabled. The user name for authentication can be included in the security configuration, for example, supAdmin@admin. If a security configuration is provided, the server looks for the application connection template according to both the appId and security configuration. If a security configuration is not provided, the server looks for the unique application connection template according to the appId. If there are multiple templates with different security configurations for the same appId, the server reports an exception, as it does not know which template should be used to authenticate the user.

- **Logical Role** – the logical role that users must belong to in order to access the application.
- **Template Priority** – the priority in which the application connection template will be used by the application, if the application has more than one application connection template associated with it.

See also

- *Creating Application Connection Templates* on page 327

Android Push Notification Properties

Android push notification properties allow Android users to install messaging client software on their devices.

Property	Description
Enabled	Enables Google Cloud Messaging (GCM) push notifications to the device if the device is offline. This feature sends a push notification over an IP connection only long enough to complete the Send/Receive data exchange. Android Push notifications overcome issues with always-on connectivity and battery life consumption over wireless networks. Acceptable values: true (enabled) and false (disabled). If this setting is false, all other related settings are ignored.
Registration ID	The registration ID that the device acquires from Google during GCM registration.
Sender ID	GCM sender ID used by SAP Mobile Server to send notifications. Used by the client to register for GCM.

Apple Push Notification Properties

Apple push notification properties allow iOS users to install client software on their devices.

- **APNS Device Token** – the Apple push notification service token. An application must register with Apple push notification service for the iOS to receive remote notifications sent by the application's provider. After the device is registered for push properly, this should contain a valid device token. See the iOS developer documentation.
- **Alert Message** – the message that appears on the client device when alerts are enabled.
Default: `New items available`.
- **Sounds** – indicates if a sound is made when a notification is received. The sound files must reside in the main bundle of the client application. Because custom alert sounds are played by the iOS system-sound facility, they must be in one of the supported audio data formats. See the iOS developer documentation.

Acceptable values: true and false.

Default: true

- **Badges** – the badge of the application icon.

Acceptable values: true and false

Default: true

- **Alerts** – the iOS standard alert. Acceptable values: true and false. Default: true.
- **Enabled** – indicates if push notification using APNs is enabled or not.

Acceptable values: true and false.

Default: true

See also

- *Configuring Messaging Subscription Settings* on page 359

BlackBerry Push Notification Properties

BlackBerry push notification properties enable the server to send notifications to BlackBerry devices using Blackberry Enterprise Server (BES).

Property	Description
Enabled	Enables notifications to the device if the device is offline. This feature sends a push notification over an IP connection only long enough to complete the Send/Receive data exchange. BlackBerry Push notifications overcome issues with always-on connectivity and battery life consumption over wireless networks. Acceptable values: true (enabled) and false (disabled). If this setting is false, all other related settings are ignored. Default: true

Property	Description
BES Push Listener Port	The listener port for BES notifications. The port is discovered and set by the client, and is read-only on the server.
Device PIN	Every Blackberry device has a unique permanent PIN. During initial connection and settings exchange, the device sends this information to the server. SAP Mobile Server uses this PIN to address the device when sending notifications, by sending messages through the BES/MDS using an address such as: Device="Device PIN" + Port="Push Listener port". Default: 0

Capabilities Properties

You cannot update or modify the capabilities properties. These properties are set by the client application and determine what properties can be set for the application connection.

- **Application Supports Client Callable Components** – Supports tracing, package users, cloning, and reregistering for application connections.
- **Application Supports Hybrid App** – Supports assigning Hybrid Apps to an application connection.
- **Application Supports Password Policy** – Supports updating the password policy for an application connection.

Connection Properties

Connection properties define the connection information for a client application so it can locate the appropriate SAP Mobile Server synchronization service.

Typically, production client applications connect to the synchronization server via Relay Server or some other third-party intermediary reverse proxy server. In those cases, the settings for the synchronization host, port, and protocol need to use Relay Server property values. For more information on how these properties are used in a synchronization environment, see *Replication in System Administration*.

- **Activation Code** – (not applicable to replication clients) the original code sent to the user in the activation e-mail. Can contain only letters A – Z (uppercase or lowercase), numbers 0 – 9, or a combination of both. Acceptable range: 1 to 10 characters.
- **Farm ID** – a string associated with the Relay Server farm ID. Can contain only letters A – Z (uppercase or lowercase), numbers 0 – 9, or a combination of both. Default: 0.
- **Server Name** – the DNS name or IP address of the SAP Mobile Server, such as "myserver.mycompany.com". If using Relay Server, the server name is the IP address or fully qualified name of the Relay Server host.
- **Server Port** – the port used for messaging connections between the device and SAP Mobile Server. If using Relay Server, this is the Relay Server port. Default: 5001.
- **Synchronization Server Host** – the server host name used for synchronization.
- **Synchronization Server Port** – the port used for synchronization.

- **Synchronization Server Protocol** – the synchronization protocol - HTTP or HTTPS.
- **Synchronization Server Stream Parameters** – the synchronization server stream parameters that are used to explicitly set client-specific values. After the client application successfully registers with the SAP Mobile Server, it receives the trusted certificate configured in the Server configuration (either the Secure Sync Port Public Certificate or Trusted Relay Server Certificate). If you are using Relay Server, ensure the Trusted Relay Server Certificate property is configured to point to a file that has the server's public certificate.

You can configure these parameters as one or more *name=value* entries.

- **trusted_certificates** – the file containing trusted root certificate file.
- **certificate_name** – the name of the certificate, which is used to verify certificate.
- **certificate_unit** – the unit, which is used to verify certificate.
- **certificate_company** – the name of the company issuing the certificate, which is used to verify certificate.

For more information about certificates, see *Security*.

- **Synchronization Server URL Suffix** – the server URL suffix. For Relay Server, suffixes vary depending on the Web Server used. For example, */cli/iarelayserver/FarmName* for Apache, or *ias_relay_server/client/rs_client.dll/FarmName* for IIS.
- **Use HTTPS** – forces clients to use HTTPS protocol for the connection.

See also

- *Configuring Messaging Subscription Settings* on page 359

Custom Settings

Define one of four available custom strings that are retained during reregistration and cloning.

Change the property name and value according to the custom setting you require. The custom settings can be of variable length, with no practical limit imposed on the values. You can use these properties to either manually control or automate how Hybrid App-related messages are processed:

- **Manual control** – an administrator can store an employee title in one of the custom fields. This allows employees of a specific title to respond to a particular message.
- **Automated** – a developer stores the primary key of a back-end database using a custom setting. This key allows the database to process messages based on messaging device ID.

See also

- *Configuring Messaging Subscription Settings* on page 359

Device Advanced Properties

Advanced properties set specific behavior for messaging devices.

- **Relay Server URL Suffix** – the URL suffix to be used when the device client is connecting through Relay Server. The suffix you set depends on whether Relay Server is installed on IIS or Apache. For IIS, this path is relative. Acceptable values include:
 - For IIS – use `/ias_relay_server/client/rs_client.dll`.
 - For Apache – use `/cli/iasrelayserver`.

Note: The value used in the client application connection for the URL suffix must match what the administrator configures in the URL suffix. Otherwise, the connection fails. Use the Diagnostic Tool command line utility to test these values. See *Diagnostic Tool Command Line Utility (diagtool.exe) Reference* in *System Administration*.

- **Allow Roaming** – the device is allowed to connect to server while roaming. Acceptable values: true and false. Default: true.
- **Debug Trace Size** – the size of the trace log on the device (in KB). Acceptable values: 50 to 10,000. Default: 50.
- **Debug Trace Level** – the amount of detail to record to the device log. Acceptable values: 1 to 5, where 5 has the most level of detail and 1 the least. Default: 1.
 - 1: Basic information, including errors
 - 2: Some additional details beyond basic
 - 3: Medium amount of information logged
 - 4: Maximum tracing of debugging and timing information
 - 5: Maximum tracing of debugging and timing information (currently same as level 4)
- **Device Log Items** – the number of items persisted in the device status log. Acceptable values: 5 to 100. Default: 50.
- **Keep Alive (sec)** – the Keep Alive frequency used to maintain the wireless connection, in seconds. Acceptable values: 30 to 1800. Default: 240.
- **Compression Enabled** – indicates whether the client software will use compression when communicating with the server. Default: true.
- **Encryption Enabled** – indicates whether the client software will use encryption when communicating with the server. Default: false.

See also

- *Configuring Messaging Subscription Settings* on page 359

Device Info Properties

Information properties display details that identify the mobile device, including International Mobile Subscriber identity (IMSI), phone number, device subtype, and device model.

- **IMSI** – the International Mobile Subscriber identity, which is a unique number associated with all Global System for Mobile communication (GSM) and Universal Mobile

Telecommunications System (UMTS) network mobile phone users. To locate the IMSI, check the value on the SIM inside the phone.

- **Phone Number** – the phone number associated with the registered mobile device.
- **Device Subtype** – the device subtype of the messaging device. For example, if the device model is a BlackBerry, the subtype is the form factor (for example, BlackBerry Bold).
- **Model** – the manufacturer of the registered mobile device.

See also

- *Configuring Messaging Subscription Settings* on page 359

Password Policy Properties

Create a password policy for device application logins. Only passwords that meet the criteria of the policy can be used to access the sensitive artifacts secured inside a device's data vault.

You can create a password policy as part of an application connection template. Ensure your developers add enforcement code to the application's data vault.

- **Enabled** – Set this value to `True` to enable a password policy for device applications. By default, this property is set to `True`.
- **Default Password Allowed** – Set this value to `True` to allow default passwords. If a default password is allowed in the policy, developers can create the vault using with a default password, by specifying null for both the salt and password arguments. By default, this value is set to `False`
- **Expiration Days** – Sets the number of days the existing password can be used before it must be changed by the user. By default, this value is set to 0, or to never expire.
- **Has Digits | Lower | Special | Upper** – Determines what combination of characters must be used to create a password stringency requirements. The more complex the password, the more secure it is deemed to be. Set the value to `True` to enable one of these password stringency options. By default they are set to `false`.

Note: Any non-alphanumeric characters are considered to be special characters.

- **Lock Timeout** – Determines how long a successfully unlocked data vault will remain open. When the timeout expires, the vault is locked, and the user must re-enter the vault password to resume using the application. Use this property in conjunction with the Retry Limit.
- **Minimum Length** – Sets how long the password chosen by the user must be. By default, this value is set to 8.
- **Minimum Unique Characters** – Determines how many unique characters must be used in the password. By default this property is set to 0. For example, if set that the password has a minimum length of 8 characters, and the number of unique characters is also 8, then no duplicate characters can be used. In this instance a password of `Sm00the!` would fail, because two zeros were used. However, `Smo0the!` would pass because the duplication has been removed.

- **Retry Limit** – Sets the number of times an incorrect password can be retried before the data vault is deleted. A deleted vault means that the database encryption key is lost, and all data in the application is rendered irretrievable. As a result the application becomes unusable. By default this value is set to 20.

Proxy Properties

(Applies only to Online Data Proxy) Proxy properties display details that identify the application and push endpoints.

- **Application Endpoint** – the URL pointing to the EIS.
- **Push Endpoint** – the SAP SAP Mobile Platform URL used by EIS to forward notifications.

Note: The application end point should be whitelisted only once as a proxy connection. The proxy connection name should be same as the application ID, if an application is registered to be used for referencing the EIS service end point.

Security Settings

Security settings display the device security configuration.

- **E2E Encryption Enabled** – indicate whether end-to-end encryption is enabled or not: true indicates encryption is enabled; false indicates encryption is disabled.
- **E2E Encryption Type** – use RSA as the asymmetric cipher used for key exchange for end-to-end encryption.
- **TLS Type** – use RSA as the TLS type for device to SAP Mobile Server communication.

Note: These settings are visible, but not in use by client (replication native application) at this time.

User Registration

User registration specifies details of the activation code that is sent to a user to manually activate an application on the device.

- **Activation Code Expiration (Hours)** – indicates how long an activation code is valid. The default is 72 hours.
- **Activation Code Length** – indicates the length of the activation code, as in number of alphanumeric characters. The default is 3.

Deploy

Deployment is a routine administration task that manages the life cycle of a mobile business object (MBO) package on the SAP Mobile Server. Deployment makes a package available to the runtime environment, so that it can be administered or accessed by client devices.

Deployment is similar to, but not the same as, exporting and importing packages between multiple cluster environments.

SAP Mobile Platform supports the development and subsequent deployment of:

- Package archives
- Hybrid App archives

Depending on the package type, the deployment steps can vary.

MBO Packages

MBO packages are collections of MBOs that are related by application use and authorship and grouped according to maintenance or distribution. Packages are initially created by developers, but are deployed and maintained on a production SAP Mobile Server by administrators.

Administrators cannot change the name of a package if one has been defined by the development team. If the development team has changed the version number of an existing package due to an upwardly incompatible change, it creates a new package when it is deployed. In this case, leave both versions of the package running until every one of the remote client applications has been upgraded to the latest version; only then should you delete the old package version. For more information about upwardly incompatible changes, see *Supported Changes for Update Deployment* in *SAP Mobile WorkSpace - Mobile Business Object Development*.

Although each mobile business object (MBO) type has unique properties and data sources, MBOs within a package used by an application may be of different types.

Note: You must deploy a package before you can configure or manage it. Package administration tasks vary, depending on the type of package you deploy.

See also

- *DOE-C Packages* on page 364
- *Hybrid App Packages* on page 367
- *Agency Applications* on page 380

Deploying MBO Packages

Deploying is the process whereby whole or part of a mobile package is loaded onto an SAP Mobile Server as one or more deployment units. SAP Mobile Server can then make these units accessible to users via a client application that is installed on a mobile device. There are different methods you can use to deploy an MBO package to SAP Mobile Server.

Deploy packages to SAP Mobile Server to test and refine configuration settings in a development environment.

After a package has been deployed, configured, tested, and refined in a development SAP Mobile Server environment, it can be transported to a QA or production server environment using export and import. This ensures that the package configuration settings on the source server are moved to the target server.

Deploying an MBO Package with the Deploy Wizard

Use the Deploy wizard to deploy an existing MBO package or to create an MBO package from a deployment unit file.

Prerequisites

If your developers have created a custom filter for the mobile business object you are deploying, you must copy class files to the primary server before you deploy the package that uses those filters. If you copy the filters to a secondary server by mistake, they are deleted when you deploy the package to the primary server.

To locate the name of the primary server, look at the server list in SAP Control Center. Servers are listed according to their cluster role (that is, primary or secondary servers). SAP Control Center automatically identifies the primary server and lists it first, followed by secondary servers.

Task

Because the deployment unit file contains package name and other information, you do not need to select a package from the list of available packages; SAP Mobile Server creates the package automatically according to what has been defined in the deployment file.

Note: If the connection properties of an MBO or operation use credentials that have been customized manually by a developer, the back-end data sources connection properties of these MBOs and operations cannot be updated by an administrator when using the Deploy Wizard.

Instead, the administrator can update these properties after the MBO or operation is deployed to an SAP Mobile Server. When the server connection is created on SAP Mobile Server, the administrator can then change connection properties in SAP Control Center by clicking on the Connections node in the left navigation pane.

1. In the left navigation pane, expand the **Domains** folder.

2. Choose a domain name, then select **Packages**.
3. In the right administration pane, click the **General** tab.
4. Click **Deploy**.
Follow the instructions in the wizard to configure a package so it can be deployed.

See also

- *Transporting Packages Between Environments with Export and Import* on page 346

Configuring Deployment Properties

Set the properties for packages being deployed on SAP Mobile Server.

A deployment file and an optional deployment descriptor file are created by developers in SAP Mobile WorkSpace. These files are typically delivered for deployment to a production version of the SAP Mobile Server by an administrator.

Note: If the package is deployed to the primary server, it is cluster-level operation.

1. When the **Deploy** wizard loads, click **Next**.
2. Review the **Deployment File** name, or click **Browse** to navigate to the appropriate file.
You can select either a single deployment unit (. XML) or an archive file (. JAR).

The name of the **Package Name** appears. If this package does not already exist, the wizard displays a message that indicates the new package will be created. The name cannot exceed 64 characters or include any periods (".").

3. Select a **Deployment Mode**.

The deployment mode determines how the deployment process handles the objects in a deployment unit and package. The value you choose depends on whether or not a package of the same name exists on SAP Mobile Server. Allowed values are:

- **UPDATE** – (default) updates the target package with updated objects. After deployment, objects in the server's package with the same name as those being deployed are updated.

Use update where existing deployments and client applications need to continue to work. After a successful (or failed) Update deployment, existing client applications previously synchronized with SAP Mobile Server continue to function: existing surrogate keys do not change, all operations are fully functional, and so on. No data loss on the devices occur, and the existing client can synchronize and upload operations as if the Update deployment never occurred.

- **VERIFY** – do not deploy package. Only reports what, if any, errors would occur if you were to deploy the package using **UPDATE** deploy mode.

Verify ensures that any existing client is still able to operate fully after a deployment. A limitation for message-based synchronization clients is that the old client could receive unnecessary data that is ignored on the device.

If the deployment mode is specified both in the descriptor file and the Deploy wizard, the Deploy wizard Deployment Mode option overrides the deployment mode specified in the descriptor file.

4. Browse and select an optional deployment **Descriptor File**.
5. Click **Next**.
6. Select a **Domain** to deploy the package to.
7. Select a **Security Configuration** for the package.
8. Click **Next**.

The Configure Role Mapping page appears.

Deployment Archives

An archive is produced after a developer creates a package profile and executes a build on a package. This archive can only be created in the Eclipse edition of SAP Mobile WorkSpace.

In SAP Mobile WorkSpace, a developer executes a build process so that it creates a `.jar` archive file, which contains both a deployment unit and a corresponding descriptor file. A deployment archive can be delivered to an administrator for deployment to a production version of the SAP Mobile Server.

Deployment Descriptors

A deployment descriptor is an XML file that captures changes to the deployment unit during deployment. Those changes are then used when a package is redeployed.

A deployment descriptor is not required to deploy the deployment unit.

A deployment descriptor is created either after a developer creates a package profile and executes a build on a package, or when the developer deploys a package to a development edition server from SAP Mobile WorkSpace. The file contains this information:

- The deployment mode
- The target package that descriptor applies to
- The endpoint information that overrides specific endpoints defined for MBOs or operations in the deployment unit
- The domain and named security that the package applied to
- Role mappings

This information is specific to each deployment unit; therefore, you cannot apply a descriptor from another package to a deployment unit.

Mapping Roles for a Package

SAP Mobile Platform uses a role mapper to map logical and physical roles during an access control check. This allows developers to create applications that incorporate a logical access control policy. When the application is deployed, a security administrator can work with the developer to understand what the logical roles in the application were intended to do and map these logical roles to physical roles that exist in the real security system. Role mappings

performed at the package level override the mappings set at the global level. Package-level role mappings apply to all packages that use the same security configuration, even if the package is deployed in multiple domains. You can set the mapping state either when managing roles, or after package deployment.

Prerequisites

Unwired Platform cannot query all enterprise security servers directly; to perform authentication successfully, know the physical roles that are required.

Task

1. For package-specific role mapping, select and deploy an available package. Follow the wizard prompts until you reach the Configure Role Mapping page for the target package.
2. Select a logical role and select one of the following in the adjacent list:

State	Description
AUTO	To map the logical role to a physical role of the same name.
NONE	To disable the logical role, which means that the logical role is not authorized.
MAP	To manually map the logical role when the physical and logical role names do not match. See <i>Mapping a Physical Role Manually</i> .

3. Click **Next**.

The Server Connection page appears.

Deployment-time role mapping is done at the package level. Once the package is deployed, you can change the role mapping by going to the Role Mapping tab for the desired package.

Mapping a Physical Role Manually

Use the Role Mappings dialog to manually map required physical roles for a logical role when physical and logical role names do not match. If names do not match, the AUTO mapping state does not work.

Prerequisites

SAP Mobile Platform cannot query all supported enterprise security servers directly; for successful authentication, you must know the physical roles your back-end systems require.

Task

You can map a logical role to one or more physical roles. You can also map multiple logical roles to the same physical role. If a role does not exist, you can also add or delete names as needed.

1. Review the list of existing physical role names that you can map to the logical role you have selected. If the list retrieved is too long to locate the name quickly, either:
 - Click the banner of Available Roles list to sort names alphanumerically.
 - Start typing characters in the box, then click the Search button to filter the available list.
2. If a role that you require still does not appear, enter the **Role name** and click the + button. The role name appears in the **Available roles** list with an asterisk (*). This asterisk indicates that an available role was added by an administrator, not a developer.
3. To remove a role you no longer require from the **Available roles** list, select the name and click the x button adjacent to the **Role name** field. The role is removed and can no longer be mapped to a logical role.
4. To map a logical role that appears in the text area of the Role Mappings dialog to a physical role:
 - a) Select one or more **Available roles**.
 - b) Click **Add**.
5. To unmap a role:
 - a) Select one or more **Mapped roles**.
 - b) Click Remove.

The roles are returned to the **Available roles** list.
6. Click **OK** to save these changes.

Once a logical role has been manually mapped, the mapping state changes to MAPPED. The roles you have mapped appear in the active Physical Roles cell for either a package-specific or server-wide role mappings table.

Mapping State Reference

The mapping state determines the authorization behavior for a logical name instance.

State	Description
AUTO	Map the logical role to a physical role of the same name. The logical role and the physical role must match, otherwise, authorization fails.
NONE	Disable the logical role, which means that the logical role is not authorized. This mapping state prohibits anyone from accessing the resource (MBO or Operation). Carefully consider potential consequences before using this option.
MAPPED	A state that is applied after you have actively mapped the logical role to one or more physical roles. Click the cell adjacent to the logical role name and scroll to the bottom of the list to see the list of mapped physical roles.

Updating Server Connection Properties

Configure the production version of server connection properties. Typically, the endpoint and role mapping a developer would use are not the same for a production system. Administrators must reset these properties accordingly.

This step allows you to change connection profiles used for development (design time) to an appropriate server-side connection. For example, your development environment might permit access to certain systems that the SAP Mobile Server prohibits.

Note: If the connection properties of an MBO or operation use credentials that have been customized manually by a developer, the back-end data sources connection properties of these MBOs and operations cannot be updated by an administrator when using the Deploy Wizard.

1. On the left pane, select an **MBO** or an **operation**.
2. Review the connection properties.
3. Depending on how the properties are configured, choose an appropriate option:
 - If the properties are configured by the developer as a set of connection properties, you can edit properties as needed. To edit a property, click a field in the **Value** column and change the value as required.
 - If the properties are configured as an endpoint, choose an existing server connection to use. You cannot alter any of these properties.

If the package contains MBOs that belong to an EIS managed cache group, you can set or change the Web Service endpoint. This endpoint is used by SAP Mobile Server to send a notification to inform the EIS of the availability of updates. For more information, see *EIS Managed Cache Group Policy* in *SAP Mobile WorkSpace - Mobile Business Object Development*.
4. When connection properties change for an MBO, you can apply the changes to operations as well by clicking the corresponding check-box at the bottom of the table. When connection properties change for an operation, you can apply the changes to the MBO or to other operations by clicking the corresponding check-box at the bottom of the table.
5. Click **Next**.

The connection properties are updated. The changes you make are displayed in the Summary page in the Endpoint Updates section.

Reviewing the Deployment Summary

Review the properties you have supplied before deploying the package to SAP Mobile Server. This allows you to change errors before making the deployment units available via the package created for that purpose.

1. Review all three sections of the deployment summary. If anything is incorrect, click **Back** and correct errors.

2. To create a deployment descriptor:

- a) Click **Create deployment descriptor**.
- b) Browse to the location you want to save the file and change the default name if required.
- c) Click **Save**.

You can use the deployment descriptor to redeploy the deployment unit without having to repeat these steps in the deployment wizard.

3. Click **Finish** to deploy the file and create the package on SAP Mobile Server.

Deploying to the server may take some time to complete. However, a status message appears above the **General** tab indicating the success or failure of the attempt.

Transporting Packages Between Environments with Export and Import

If you are trying to transfer deployment packages among disparate server environments, you can export packages from one server environment before importing them into another one. This sequence of events is typically performed by administrators when transporting packages from a development local server environment to a test network environment, or from a test or pilot environment to production cluster environment.

See also

- *Deploying an MBO Package with the Deploy Wizard* on page 340

Exporting MBO Packages

Export an MBO package to bundle one or more MBOs and package options to create a new instance of a deployment archive. Use the deployment archive to transport the package between SAP Mobile Servers.

Prerequisites

Before beginning, review import requirements and best practices.

Task

1. In the left navigation pane of SAP Control Center, expand **Domains** > **Domain name** > **Packages**.
2. In the right administration pane, select the box adjacent to the name of the package and click **Export**.
3. Click **Next**.
4. Click **Finish**.
5. Select the file system target for the exported contents and click **Save**.

Note: Ensure that you do not hide the file type extension when you name the export archive; otherwise, the *.zip extension becomes invisible, which adversely affects the outcome of the export process.

A status message indicates the success or failure of the export transaction. If the transaction succeeds, a ZIP file is created in the location you specified. You can then import this file on another SAP Mobile Server.

Next

Deliver the file to the appropriate person, or deploy or transport the exported package to the appropriate server.

See also

- *Importing MBO Packages* on page 347
- *Import Requirements and Best Practices* on page 347

Importing MBO Packages

Import MBO packages after they have been exported from another SAP Mobile Server.

Prerequisites

Review import requirements and best practices before beginning.

Task

1. In the left navigation pane of SAP Control Center, expand **Domains** > *<domainName>* > **Packages**.
2. In the right administration pane, click **Import**
3. Click **Browse** to navigate to the file.
4. Click **Import**.
5. After the import is complete, check the import details and click **OK**.

See also

- *Exporting MBO Packages* on page 346
- *Import Requirements and Best Practices* on page 347

Import Requirements and Best Practices

Import is typically used to move a package or application from a development environment to a test environment, and after testing to a production environment.

MBO Package Imports

- **Domain requirements** – all server connections and security configurations referenced by the MBO package must exist in the target domain.
- **Versioning recommendations** – if a developer has updated the package version number:

1. (Required) Verify that this new package version is added to the application.
2. (Recommended) Whenever possible, use the update instead of import. Otherwise, delete the existing package first, to remove all runtime data for the package including cached data, registered subscriptions, subscription templates, client log, MBO and operation histories, and registered package users. Delete these items only after serious consideration.

Application Imports

Make sure the target system has resources that match those referenced in the export archive file:

- Domains, security configurations, logical roles assigned to application connection templates, proxy endpoint connections (used by ODP applications)
- If MBO packages are included, the domains, security configurations and connections referenced by the MBO package archives

Hybrid App Imports

If MBO packages are referenced in the export archive file, make sure that the MBO packages have already been deployed to the target system.

Note: If the Hybrid App has matching rules, all matching rule search expressions are imported as regular expression types. Other expression types such as `Begins with` or `Equals` are imported as `Regular expression`.

See also

- *Exporting MBO Packages* on page 346
- *Importing MBO Packages* on page 347

MBO Package Management

There are common package management activities used by all package types: replication, messaging, unified, DOE-C, or ODATA.

These activities include:

See also

- *DOE-C Packages* on page 364

Deleting a Package

Delete a package when you want to permanently remove all elements deployed on an SAP Mobile Server. If you do not want to permanently prohibit access by removing these files, consider disabling the package instead.

Deleting a package removes all runtime data for the package including cached data, registered subscriptions, subscription templates, client log, MBO/Operation history, and registered

package users. All clients must re-sync as a result. Therefore, deletion action should be taken after careful consideration.

1. In the left pane, click the SAP Mobile Server you are currently logged in to.
This expands the list of SAP Mobile Platform components you can manage, provided that you have the correct permissions to do so.
2. Click **Domains** > <domainName> > **Packages**.
3. To delete a package from the this list, select one or more packages and click **Delete**.
A confirm dialog box appears.
4. Click **Yes** to confirm the deletion.

The package is removed from SAP Mobile Server.

Enabling and Disabling a Package

Enable or disable a package to allow or prohibit device access to the package. Disabled packages are still available to SAP Control Center for SAP Mobile Server. By default, all packages are enabled.

When you disable a package, the server unloads all of its elements from memory. Disabling a package prevents the SAP Mobile Server from loading that package at start-up.

Note: You cannot disable a SAP® Data Orchestration Engine Connector (DOE-C) package.

1. In the left navigation pane, click **Domains** > <domainName> > **Packages**.
2. In the right administration pane, select the **General** tab.
3. Select the box adjacent to the package you want to enable or disable.
You can select more than one package to apply the same change to multiple components.
4. Depending on the current status of the package, perform one of:
 - Enable – if the package listed shows a status of disabled, click **Enable**.
 - Disable – if the package listed shows a status of enabled, click **Disable**. The package remains disabled until you or another administrator enables it; restarting SAP Mobile Server does not enable the package.

Configuring Synchronization Groups

Configure a synchronization group when a collection of MBOs must be synchronized at the same time.

The synchronization group defines both the logical unit of synchronization, and generation frequency for data notifications. The latter influences the synchronization frequency for replication synchronization applications, which usually need to perform a synchronization event to download the changes associated with the data notification event.

The synchronization group properties jointly determine the frequency with which data changes are detected, and notifications are generated. The actual data notification delivery to the application is determined by this equation:

```
delivery of data notification = cache schedule repeat value or data  
change notification arrival time + synchronization group change  
detection interval value + device's subscription push interval  
setting
```

There may be an additional 10 second delay, because the internal thread for performing the change detection runs every 10 seconds.

1. In the left navigation pane, expand the **Domains** > *<domainName>* > **Packages** folder and select the package containing the sync group you want to configure.
2. In the right administration pane, click the **Synchronization Group** tab.
3. Select the desired sync group and click **Properties**.
4. Select a change detection interval. This value determines how frequently SAP Mobile Server looks for changes to MBOs, and generates push notifications.

If you set the **Change detection interval** to 0, the change detection task is disabled and the server does not push changed messages or data to the client device.
5. Click **OK**.

Configuring a Cache Group

Select and configure a cache group. The cache is part of the SAP Mobile Server cache database (CDB) that is used to store data that is uploaded and downloaded from EIS servers and mobile clients during synchronization.

Cache group configuration differs depending on whether the cache group is defined as "on demand" or "scheduled" during development.

MBO Data in the CDB

A data cache is a copy of MBO data that is stored in a specific area of the cache database (CDB). It is used as the data repository for replication and messaging MBOs that are deployed to SAP Mobile Server. "CDB" and "cache" and "MBO data" can sometimes be used interchangeably, even though the CDB includes runtime data as well.

When cache data is updated (either with an on-demand or scheduled cache refresh), the remote client database eventually retrieves the updated data from the server's copy of MBO data in the CDB by synchronization.

By giving applications a normalized and uniform view of corporate data, organizations can:

- Lower the barrier to data behind corporate firewalls
- Support development of mobile applications that interact with multiple enterprise back-ends
- Reduce back-end load caused by device client requests

Configuring On Demand Cache Group Properties

Specify the duration of cache data validity by configuring SAP Mobile Server updates to mobile business object (MBO) data for an on demand cache group.

Prerequisites

You can configure on demand updates for a cache only if the developer enables the cache group as "on demand" during development. Otherwise, the Cache tab is not configurable in the Cache Group Properties dialog.

Task

1. In the left navigation pane, expand the **> Domains > <domainName> > Packages** folder, and select the package for which you want to configure cache settings.
2. In the right administration pane, click the **Cache Group** tab.
3. Select the cache group you want to configure and click **Properties**.
4. In the **Cache Properties** dialog, enter an expiry for the **Cache Interval** in seconds, minutes, or hours.

The cache interval determines how frequently SAP Mobile Server updates the cache database with changes to enterprise data. See *On Demand Cache Refreshes* in the list of links below.

5. Click **OK**.

On Demand Cache Refreshes

Cache groups designated as "on demand" during development use cache intervals to balance how frequently the object updates enterprise data with the amount of network traffic required to maintain that data.

SAP Mobile Server keeps a local copy of enterprise data in the cache database (CDB), and uses an intricate mechanism to manage updates between the CDB and the EIS servers. When data is updated, the remote client database eventually gets updated data from this local copy in the CDB. The caching mechanism allows MBOs to retrieve updated data even if back-end servers fail.

You must choose an appropriate cache interval for your system, since this value determines how frequently the CDB is updated with data from the EIS. The cache interval must be configured according to business needs. A higher value for the cache may retain stale data, however, a lower value increases the backend EIS load and may impede the client application's performance, because SAP Mobile Server queries the back-end information servers more frequently to look for changes and possibly update the CDB copy.

Frequent queries typically put a higher load on the servers, as well as use more network bandwidth on the server side. While the cache interval does not affect the bandwidth required between the synchronization server and device client applications, nor the performance

characteristics of the client applications, the interval you choose can delay synchronization if SAP Mobile Server must first update many records in the CDB.

For example, if the cache interval is 0, each time a client application synchronizes, there is a pause while the SAP Mobile Server rereads data from the EIS and updates the CDB. If, however, the cache interval is greater than 0, then the wait time depends on how long ago the data was refreshed. If the synchronization falls within a recent cache update, synchronization is almost immediate.

Configuring Scheduled Cache Group Properties

Specify the duration of cache data validity by configuring SAP Mobile Server updates to mobile business object (MBO) data for a scheduled cache group.

Prerequisites

You can configure a schedule refresh for a cache only if the developer enables the cache group as "scheduled" during development. Otherwise, the Schedule tab is not configurable in the Cache Group Properties dialog.

Task

1. In the left navigation pane, expand the **Domains** > *<domainName>* > **Packages** folder and select the package for which you want to configure cache settings.
2. In the right administration pane, click the **Cache Group** tab.
3. Select the cache group you want to configure and click **Properties**.
4. In the **Schedule** tab of the Cache Properties dialog, set the frequency of the refresh by selecting an appropriate **Schedule Repeat**: hourly, daily or custom.

This property determines what other schedule properties you must configure. Each option is documented in a separate topic which further discusses the details for each frequency type. For more details, see the corresponding topic.

Scheduling an Hourly or Daily Refresh

Scheduling an hourly or daily cache refresh means that information is fetched from the enterprise information server (EIS) and populated into the cache on either of these hourly or daily frequencies according to the schedule and range of time you configure.

The Schedule tab in the Cache Property dialog displays options appropriate for configuring this type of schedule.

1. Select either **Hourly** or **Daily** as the **Schedule Repeat** criteria.
2. (Optional) If you want to set a range to control which days the schedule refresh runs, configure a start date and time, an end date and time, or day of week (if applicable).
 - Select **Start Date** to set a date for which the first execution of the scheduled refresh is performed. To be more specific, you can also select **Start Time** to specify a start time.

If you do not set a start date and time, then, by default, the date and time that SAP Mobile Server starts is used.

- Select **End Date** to set a date that ends the repeating refresh transactions for a package. To be more specific, you can also select **End Time** to specify an end time. An end date and time are exclusive. This means that a refresh transaction runs up to, but does not include, the end time.

If you do not set an end date and time, then, by default, the date and time that SAP Mobile Server stops is used.

- Select **Specify Week Days** to select the days of the week that the refresh transaction runs. This means that for the days you select, the refresh runs every week on the day or days you specify. A weekday is inclusive. This means that any day you choose is included in the frequency. All others are excluded.

For example, if a schedule has a start time of 13:00 and an end time of 16:00 and repeats every hour, the refresh task would execute between 13:00 and 16:00 as determined for a custom schedule (based on the interval expiry). For an hourly or daily schedule, the refresh is triggered in that time range on an hourly or daily basis.

When the schedule expires, the automatic refresh you configured terminates, unless the end user initiates a refresh.

3. Click **OK**.

See also

- *Scheduling a Custom Refresh* on page 353
- *Scheduled Cache Refreshes* on page 354

Scheduling a Custom Refresh

Scheduling a custom cache refresh is the most flexible of all cache refresh schedules. This means that information is fetched from the enterprise information system (EIS) according to the schedule repeat interval you specify.

The Schedule tab in the Cache Property window displays options appropriate for configuring this type of schedule.

1. Select **Custom** as the schedule repeat criteria.
2. Specify a repeat **Interval**, in hours, minutes or seconds, to determine how often the cache refresh occurs.

This interval determines how frequently SAP Mobile Server updates the cache database with changes to enterprise data. The data is held by the cache for the duration of the specified interval.

3. (Optional) To set a range to control which days the schedule refresh runs, configure a start date and time, end date and time, or day of week (if applicable).
 - Select **Start Date** to set a date for which the first execution of the scheduled refresh is performed. To be more specific, you can also select **Start Time** to specify a start time.

In this case, the refresh cannot begin until a given time on a given day has been reached. A start date and time are inclusive.

If you do not set a start date and time, then, by default, the date and time that SAP Mobile Server starts is used.

- Select **End Date** to set a date that ends the repeating refresh transactions for a package. To be more specific, you can also select **End Time** to specify an end time. An end date and time are exclusive. This means that a refresh transaction runs up to, but does not include, the end time. For example, if a schedule has a start time of 13:00 and an end time of 16:00 and repeats every hour, it runs at 13:00, 14:00, and 15:00, but not at 16:00.

If you do not set an end date and time, then, by default, the date and time that SAP Mobile Server stops is used.

- Select **Specify Week Days** to select the days of the week that the refresh transaction runs. This means that for the days you select, the refresh runs every week on the day or days you specify. A weekday is inclusive. This means that any day you choose is included in the frequency. All others are excluded.

When the schedule expires, the automatic refresh you configured terminates, unless the end user initiates a refresh.

4. Click **OK**.

See also

- *Scheduling an Hourly or Daily Refresh* on page 352
- *Scheduled Cache Refreshes* on page 354

Scheduled Cache Refreshes

A schedule-driven cache refresh is a background task that runs between a configured start and endpoint at scheduled intervals during normal server operation.

A schedule-driven cache refresh defines a contract between SAP Mobile Server and back-end information servers. Normally, data is retrieved from a server (for example, a database, and an SAP repository, or a Web service) when a device user synchronizes. If the administrator wants the data to be preloaded, he or she configures the SAP Mobile Server repeat interval to expedite data updates on the device.

Two properties configure the cache refresh schedule, which is used with a subscription to synchronize data for mobile business objects (MBOs).

- **Schedule repeat** – determines the time frame when data is refreshed. If you set up a schedule to repeatedly refresh data, information is always refreshed. Set the schedule to meet business application requirements for data consistency.

As an administrator, you may also use a schedule repeat to look for data changes and alert subscribed clients to synchronize when there are changes. Keep in mind, however, that the actual detection of changes and sending of data (for messaging payloads) or notifications (for replication payloads) depends on the:

- **Change detection interval** property of the synchronization group for the package.
- **Notification threshold** property of subscriptions for replication payloads.
- Push related device settings for messaging payloads.
- Repeat interval – determines how often SAP Mobile Server updates the cache with changes to backend data.

See also

- *Scheduling an Hourly or Daily Refresh* on page 352
- *Scheduling a Custom Refresh* on page 353

Online Refresh Policy

MBOs that use an Online refresh policy indicates that the MBOs are to be used only in Hybrid App applications where access to real-time enterprise information system (EIS) data is required (cache validity is zero).

Data is valid in the SAP Mobile Server cache only until delivery and immediately invalid. You cannot modify the cache or schedule of the Online policy. Expired data is purged from the cache based on a schedule at the domain level.

DCN Refresh Policy

The cache refresh and schedule options are disabled for DCN (data change notification) policy, since data never expires and is not refreshed based on client demand or a schedule.

Cache data does not expire until a cache invalidate operation is invoked or a data change notification request is received from the enterprise information system (EIS).

EIS Managed Refresh Policy

MBOs that use an EIS managed refresh policy indicates that the enterprise information system (EIS) initiates the connection to SAP Mobile Server to retrieve updates.

When deploying a package that contains MBOs that belong to an EIS managed cache group, you can map the Web Service endpoint to a server endpoint or create a new one based on the endpoint set on the EIS cache group before deployment. This endpoint is used by SAP Mobile Server to send a notification to inform the EIS of the availability of updates.

Purging the Synchronization Cache Manually

You can manually purge the synchronization at the package level.

1. In the left navigation pane, expand the **Domains** folder and select a domain.
2. Navigate to the **Packages** folder, and select a package.
3. In the right pane, select the **Cache Group** tab.
4. On the **Cache Group** tab, click **Purge**.
5. Click **OK** to purge immediately based on your selections.

Assigning Package-Level Security

Assign security configurations at the package level to override those set at the domain-level.

1. *Selecting a Security Configuration for a Package*

Designate a security configuration for a package in SAP Control Center. This is a required step during package deployment, but you can later change the security configuration.

Selecting a Security Configuration for a Package

Designate a security configuration for a package in SAP Control Center. This is a required step during package deployment, but you can later change the security configuration.

The administrator must create a security configuration in the cluster and assign it to the domain where the package is deployed before the deployer can assign the security configuration to the package.

1. In the left navigation pane, expand the **Domains** > *<domainName>* > **Packages** folder, and select the package to configure.
2. In the right administration pane, click the **Settings** tab.
3. Select a security configuration.

The security profiles that appear in this list have been created by a platform administrator and assigned to the domain.

4. Click **Save**.

Viewing Asynchronous Operation Replays

View pending asynchronous operation replays for unified or replication packages already deployed to SAP Mobile Server.

Prerequisites

Asynchronous operation replays are only available if the feature has been enabled as part of the cluster properties. See *Configuring Asynchronous Operation Replay Queue Count*.

Task

1. In the left navigation pane, expand the **Domains** > *<domainName>* > **Packages** folder and select the deployed package.
2. In the right administration view, click **the Asynchronous Operation Replay** tab.
3. Review the requests displayed in the table for this package.

The number of replays are ordered in sequence along with:

- The user name that performed the operation.
- The name of the operation.
- The name of the MBO.

4. If the list is long, you can search for a particular entry:

- a) Select the field you want to search for: user, operation, or MBO name.
- b) Type the text string in the adjacent field.

This file allows the use of wildcards (that is, *). For example, if you selected a user name search, and you want to return replays for all users that start with A, you type A* as the search string.

See also

- *Configuring Asynchronous Operation Replay Queue Count* on page 49
- *Viewing Cluster Information* on page 64

Viewing and Changing Package Connection Properties

View and edit connection properties for deployed DOE-C packages.

1. In the left navigation pane, expand the **Domains** > *<domainName>* > **Packages** folder, select the package whose connection properties you want to view or change, and select the mobile business object or the operation.
2. In the right navigation pane, click the **Connection** tab.
3. If you wish to change any of the settings:
 - a. In the left navigation pane, click the **Domains** > *<domainName>* > **Connections** node.
 - b. In the right administration pane, click the **Connections** tab.
 - c. Select the checkbox for the **Connection Pool Name** that matches the name of the package whose connection properties you were just viewing.
 - d. Click **Properties**.
 - e. Make desired changes and click **Save**.

Note: For DOE-C connections, you must save the connection properties first, then test connection. For other connection types, you can test connection without saving.

- f. Click **Test Connection**.

If the connection test is not successful, see *DOE-C or Messaging Connection Test Errors* in the *Troubleshooting* guide

MBO Subscription Management

Manage subscriptions for MBO packages.

Subscription management activities include:

Configuring Package Subscriptions

Configure subscriptions and subscription templates to allow the device user to be notified when information is available, depending on the subscription properties configured for a package. Subscription templates allow you to configure predefined properties for a

synchronization group. A client's first synchronization for a specified synchronization group results in subscription creation using both the template and client-specified properties.

Creating Replication Subscription Templates

Create a subscription template to specify synchronization targets and behavior for subscribed users. A template is useful to create a set of predefined values that are used frequently. Otherwise, a subscription is still automatically created for each client upon explicit indication of interest for a device notification.

This is an optional step; it is only required if an administrator wants to establish preset subscription properties. The subscription properties can be modified from device application, but only if the **Admin lock** property is disabled.

1. In the left navigation pane, expand the **> Domains > <domainName> > Packages** folder and select the replication based sync package you want to configure.
2. In the right administration pane, click the **Subscriptions > Replication** tab.
3. From the menu bar, select **Templates**.
4. Click **New**.
5. In the New Template dialog, select settings for these options:
 - Synchronization Group – the group of MBOs that a client receives data change notifications for when data changes occur.
 - Notification Threshold – the length of time that must pass since a client's last synchronization before another notification is sent.
 - Admin Lock – (enable or disable) prevents device users from modifying the push synchronization state or sync interval value configured in the subscription. If the admin lock is disabled, the device client user can change these properties, and these changes take effect the next time the client user synchronizes the package to which the subscription applies.
 - Push – (enable or disable) if enabled, automatic server-initiated notifications are pushed to users when changes occur in the cache. If disabled, device users perform client-initiated synchronizations when they receive an outbound notification.

Note: If you intend to use push synchronization with BlackBerry devices, enable push synchronization in the BlackBerry server. See the BlackBerry server documentation for details.

6. Click **OK**.

The new subscription template appears in the list of templates.

Notifications are delivered to BlackBerry device clients using push-based notification settings and to Windows Mobile device clients using pull-based notification settings. The "poll every" setting determines the notification delivery behavior. See in SAP Control Center online help.

Configuring Replication Subscription Properties

View and configure subscription properties for replication device users subscribed to deployed synchronization packages.

1. In the left navigation pane, expand the **> Domains > <domainName> > Packages** folder and select the replication-based synchronization package you want to configure.
2. In the right navigation pane, click the **Subscriptions > Replication** tab.
3. From the menu bar, select **Devices**.
4. Check the box adjacent to a device user and click **Properties** to view these subscription properties:
 - Application Connection ID – the unique identifier for a user application connection.
 - User ID – the ID of the user associated with the application ID. The user under the domain logs should be in the format "username@securityConfiguration".
 - Package Name – the name of the package to which the subscription belongs.
 - Sync Counts – the total number of synchronizations for the subscription since the synchronization history was last cleared.
 - Notification Threshold – the length of time that must pass since a client's last synchronization before another notification is sent.
 - Last Sync Time – the date and time that the last synchronization for the subscription occurred.
 - Synchronization Group – the group of MBOs that a client receives data change notifications for when data changes occur.
 - Admin Lock – (enable or disable) prevents device users from modifying the push synchronization state or sync interval value configured in the subscription. If the admin lock is disabled, the device client user can change these properties, and these changes take effect the next time the client user synchronizes the package to which the subscription applies.
 - Push – (enable or disable) if enabled, automatic server-initiated notifications are pushed to users when changes occur in the cache. If disabled, device users perform client-initiated synchronizations when they receive an outbound notification.
5. Check the box adjacent to **Clear sync history** in order to erase stored synchronization details for the subscription.
6. If you made changes to subscription properties, click **Save**. Otherwise, click **Cancel** to return to the Subscriptions tab.

Configuring Messaging Subscription Settings

View and edit user properties for messaging packages that allow you to manage messaging between SAP Mobile Server and application users.

Deploy

1. In the left navigation pane, expand the **Domains** > **<domainName>** > **Packages** folder, and select the package you want to configure.
2. In the right navigation pane, click **Subscriptions** > **Messaging**.
3. Check the box adjacent to an application connection and click **Device Settings**.
4. Configure these property categories, as required:
 - Connection
 - Custom Settings
 - Device Advanced
 - Device Info
 - User Registration
 - Apple Push Notifications (iPhone only)
 - BlackBerry Push Notifications
 - Proxy Security Settings
 - Application Setting

See also

- *Connection Properties* on page 333
- *Custom Settings* on page 334
- *Device Advanced Properties* on page 335
- *Device Info Properties* on page 335
- *Apple Push Notification Properties* on page 332

Managing Subscriptions

Manage subscriptions to control the messages that application connections receive.

Subscription management tasks include pinging, unsubscribing, recovering, suspending, resuming, resynchronizing, and logging subscriptions. Subscription tasks vary by the package type.

These subscription management tasks apply only to the package types specified in the table below. Perform each task in the Subscriptions tab of the deployed package you are managing.

Table 22. Subscription management tasks

Subscription task	Description	Summary	Package type
Ping	Ensure that push information a user provides for a device is configured correctly. If the ping is successful, notifications and subsequent data synchronizations occur as defined by each subscription. If the ping fails, open the log and check for an incorrect host name or port number.	Select the box adjacent to the device ID, and click Ping .	UNIFIED (replication subscriptions)
Unsubscribe	Remove a subscription from SAP Mobile Server.	Select the box adjacent to the device ID, and click Unsubscribe for UNIFIED packages, messaging packages, and DOE-C packages. For Windows Mobile, the device application must include the <code>DatabaseClass.CleanAllData();</code> method for data to be unsubscribed correctly. If this method is not used, Unsubscribe could work unpredictably.	All
Recover	Reestablish a relationship between the device and SAP Mobile Server. Perform recovery under severe circumstances when a device is unable to successfully synchronize data. During subscription recovery, SAP Mobile Server purges all enterprise data on the device. It retains the device ID and subscription information so that all data can then be resynchronized and loaded onto the device.	Check the box adjacent to the subscription ID of the device, and click Recover .	UNIFIED (messaging subscriptions)

Subscription task	Description	Summary	Package type
Suspend/resume	Control the deactivation and reactivation of package subscriptions: - Suspend – temporarily block data synchronization for a device subscribed to a particular package. - Resume – reactivate a package subscription after it has been suspended.	Select the box adjacent to the subscription ID of the device, and click either Suspend or Resume .	UNIFIED (messaging subscriptions) DOE-C
Resynchronize	Reactivate subscriptions to a deployed package. If a DOE-C subscription does not respond to the SAP DOE quickly enough, the DOE may mark that subscription's queues as "blocked" and stop sending messages to the DOE-C. Resynchronize to resume communication from the DOE to the DOE-C subscription.	Check the box adjacent to the subscription ID of the device, and click ReSync .	DOE-C
Purge	Removes subscriptions that are no longer referenced by any active users.	Select the subscription, click Purge , and then select the criteria.	UNIFIED (messaging subscriptions) UNIFIED (replication subscriptions)

Purging Inactive Package Subscriptions Manually

Purge subscriptions for that have been inactive and remove them from the cache database as well as SAP Control Center's management view.

Note: Devices for which you have purged subscriptions cannot perform any operations. Only purge those subscriptions that are inactive for a long period of time.

1. In the left navigation pane, expand the **Domains** folder and select a domain.

2. Navigate to the **Packages** folder, and select a package.
3. In the right pane, select the **Subscriptions** tab.
4. Select the application payload used: replication or messaging.
5. Click **Purge**.
6. Enter a positive integer for the **Number of Inactive Days**.
7. Click **OK**. The subscriptions that match the purging criteria are physically deleted from cache database.
8. Repeat the step for the alternate payload protocol in a mixed application environment.

Reviewing MBO History

View or clear the error history of a mobile business object (MBO).

1. In the left navigation pane, expand the **Domains > <domainName> > Packages** folder and select the package that contains the MBO you want to view.
2. Select the MBO.
3. In the right administration pane, click the **History** tab.
4. To view MBO data from a specific time period, select a **Start date** and **End date** and click **Go**.
5. Review the following data for the MBO:
 - Data refresh time – the time of this data refresh's failure.
 - Failed data refresh counts – the number of failed data refreshes that occurred during the specified time period.
 - Last successful data refresh – the date and time of the most recent successful data refresh of this MBO before this refresh failure.
6. To clean MBO history data, click **Clean**.
Data is removed from the cache database.

Reviewing Operation History

View the error history of a mobile business object (MBO) create, delete, or update operation.

1. In the left navigation pane, expand the **Domains > <domainName> > Packages** folder and select the package that contains the MBO operation you want to view.
2. Expand the desired MBO and select the operation for which you want to view the error history: **create**, **delete**, or **update**.
3. In the right administration pane, click the **History** tab.
4. To view operation data from a specific time period, select a **Start date** and **End date** and click **Go**.
5. Review the following data for the operation:
 - Operation replay time – time of this operation replay's failure.

Deploy

- Number of failed operation replays– the number of failed operation replays that occurred during the specified time period.
 - Last successful operation replay – the date and time of the most recent successful replay of this operation before this operation replay failure.
6. Click **Clean** to remove operation history data.
The lines are removed from the cache database.

DOE-C Packages

Use SAP® Data Orchestration Engine Connector (DOE-C) packages if you want to be able to connect to data on an SAP back end..

DOE-C packages implement the messaging payload protocol, so package management that relate to messaging packages also apply to DOE-C packages. The activities listed here only apply to DOE-C.

See also

- *MBO Packages* on page 339
- *Hybrid App Packages* on page 367
- *Agentry Applications* on page 380
- *MBO Package Management* on page 348

Deploying and Configuring DOE-C Packages

Unlike Hybrid App or MBO packages that use SAP Control Center to deploy packages to SAP Mobile Server, you must deploy the DOE-C package to specific domain using the DOE-C command line utility (CLU). Once deployed, the DOE-C package is visible and manageable from SAP Control Center.

In an SAP Mobile Platform cluster, deploy the package to the primary SAP Mobile Server node – SAP Mobile Platform automatically replicates the package to the other nodes.

To provide failover and load balancing in an SAP Mobile Platform cluster, specify the URL of a load balancer that is capable of routing to all the SAP Mobile Server nodes in the cluster.

1. Start the command line utility console. See *Starting the Command Line Utility Console* in *System Administration*.
2. Deploy the DOE-C package. During deployment, you can set the domain and security configuration using the **setPackageSecurityConfiguration** command with -d and -sc options. After deployment, you can set the security configuration using the **setPackageSecurityConfiguration** command, or perform this task later from SAP Control Center.

See *SAP DOE Connector Command Line Utility* in the *System Administration* guide.

Next

Verify or set the security configuration for the domain or package.

Viewing and Changing Package Connection Properties

View and edit connection properties for deployed DOE-C packages.

1. In the left navigation pane, expand the **Domains > <domainName> > Packages** folder, select the package whose connection properties you want to view or change, and select the mobile business object or the operation.
2. In the right navigation pane, click the **Connection** tab.
3. If you wish to change any of the settings:
 - a. In the left navigation pane, click the **Domains > <domainName> > Connections** node.
 - b. In the right administration pane, click the **Connections** tab.
 - c. Select the checkbox for the **Connection Pool Name** that matches the name of the package whose connection properties you were just viewing.
 - d. Click **Properties**.
 - e. Make desired changes and click **Save**.

Note: For DOE-C connections, you must save the connection properties first, then test connection. For other connection types, you can test connection without saving.

- f. Click **Test Connection**.

If the connection test is not successful, see *DOE-C or Messaging Connection Test Errors* in the *Troubleshooting* guide

Setting the Bulk Load Timeout Property

The Subscribe bulk load timeout property is a package level property targeted to BlackBerry clients for initial server-side subscription operations.

Server-side subscription improves performance and is enabled on the client if the device has a secure digital (SD) memory card enabled. The timeout allows you to set an initial subscription push timeout period. If the timeout period is reached, SAP Mobile Server sends the database file to the device, whether the initial subscribe is complete or not. The timeout window signals that the device has received sufficient import messages to send the server-built database to the client.

Note: This option is only available for SAP® Data Orchestration Engine Connector (DOE-C) packages.

1. In the left navigation pane of SAP Control Center, expand the **Domains > <domainName> > Packages** folder and select the package to configure.
2. In the right administration pane, select the **Settings** tab.

3. Set the timeout value. The default value is 3600 seconds.

In addition to the timeout value, you can define the **Subscribe Bulk Load Thread Pool Size** – the maximum number of threads allocated to initial server-side subscription operations. The default value is 5. Setting the thread pool size too high can impact performance. This is a server-side setting that can be set:

- a) In the left navigation pane, select **Configuration**.
- b) In the right administration pane, select the **General** tab.
- c) From the menu bar, select **Performance**.
- d) Select **Messaging** and click **Properties**.
- e) Change the Subscribe Bulk Load Thread Pool Size value and click **Ok**.
- f) Click **Save**.
- g) Restart SAP Mobile Server for the change to take effect.

Checking and Resolving DOE-C User Failures

If the General tab of a DOE-C package displays an invalid user account error for the Error State property, you must resolve the issue by reconfiguring the username and password in the Connection Pool configured for the SAP package connection.

1. In the left pane of SAP Control Center, expand the **Domains** > *<domainName>* > **Packages** folder and click the DOE-C package name.
2. Check the Error State property in the **General** tab.
3. Validate the username and password configured, by clicking the **Connection** tab.
4. Correct the user credentials used by editing the corresponding connection pool properties:
 - a) In the navigation pane, click **Domains** > *<domainName>* > **Connections**.
 - b) In the administration pane, click the **Connections** tab.
 - c) Select the connection for the DOE-C package, then click **Properties**.
 - d) Set the username and password so that it matches the user account credentials.

Package Subscription Properties

Review information on SAP Data Orchestration Engine connector (DOE-C) package subscriptions to manage the synchronization data that device users receive.

Package subscription properties include:

- Application Connection ID – the unique identifier for a user application connection.
- User ID – the ID of the user associated with the application ID. The user under the domain logs should be in the format "username@securityConfiguration".
- Last Server Response Time – the date and time that the last outbound response was sent from SAP Mobile Server to the client.
- Client ID – the device application ID, which identifies the package database for the application.

- Application Name – the name of the device application used by the subscription.
- Status – the current status of a device. The possible values are: Running, Suspended, Pending Activation, Online, Offline, and Expired.
- Packet Dropped – the current packet dropped state of a device. The values are true or false.

Select Advanced to view these properties:

- Subscription ID – the unique identifier of the subscription.
- Logical ID – the unique identifier of a registered device that is generated and maintained by SAP Mobile Server.

Hybrid App Packages

Hybrid App packages support occasionally connected users and solve the replication and synchronization issues such users present with respect to data concurrency. Hybrid App packages are similar to other package types in that an administrator must deploy the package to SAP Mobile Server so that it can be configured and made available to client devices.

See also

- *MBO Packages* on page 339
- *DOE-C Packages* on page 364
- *Agentry Applications* on page 380
- *Deploying a Hybrid App Package with the Deploy Wizard* on page 368
- *Configuring a Hybrid App Package* on page 372
- *Enabling and Configuring the Notification Mailbox* on page 371

Deploying Hybrid App Packages

Use SAP Control Center to deploy Hybrid App packages created by developers, and to perform the configuration tasks required to make them available to application users on messaging devices. There are different methods you can use to deploy a Hybrid App package to SAP Mobile Server.

Deploy packages to SAP Mobile Server to test and refine configuration settings in a development environment.

After a package has been deployed, configured, tested, and refined in a development SAP Mobile Server environment, it can be transported to a QA or production server environment using export and import. This ensures that the package configuration settings on the source server are moved to the target server.

Deploying a Hybrid App Package with the Deploy Wizard

Use the Deploy wizard to make Hybrid App packages available on SAP Mobile Server.

If you are deploying to a target domain, replicate the value in the context variable. The domain deployment target must match the context variable defined. If the developer has used an incorrect context variable (for example, one used for testing environments), you can change the value assigned to one that is appropriate for production deployments.

1. In the left navigation pane of SAP Control Center, click **Hybrid Apps**.
2. From the **General** tab, click **Deploy**.
3. Click **Browse** to locate the Hybrid App package.
4. Select the file to upload and click **Open**.
5. Select the deployment mode:
 - New – deploys an SAP Mobile Server package and its files for the first time.
If the uploaded file does not contain an SAP Mobile Server, or an SAP Mobile Server with the same name and version is already deployed to SAP Mobile Server, you see an error message.
 - Update – installs a new SAP Mobile Server package with the original package name and assigns a new, higher version number than the existing installed SAP Mobile Server package. SAP recommends that you use this deployment mode for major new changes to the SAP Mobile Server package.

During the update operation, SAP Mobile Server:

- Acquires a list of assigned application connections from the original package.
 - Installs and assigns the package a new version number.
 - Prompts the administrator to specify application connection assignments from the acquired list of assigned application connections.
 - Preserves existing notifications.
 - Preserves the previous SAP Mobile Server package version.
- Replace – replaces an existing SAP Mobile Server package with a new package, but maintains the same name and version. SAP recommends that you use the replace deployment mode for minor changes and updates to the SAP Mobile Server package, or during initial development.

During the replace operation, SAP Mobile Server:

- Acquires a list of assigned application connections for each user of the original package.
- Uninstalls the original package.
- Installs the new package with the same name and version.
- Assigns the original application connections list to the new package, thus preserving any application connection assignments associated with the original package.

The package is added to the list of deployed packages, which are sorted by Display Name.

Next

Configure the deployed package if you want it to have a different set of properties in a production environment.

See also

- *Configuring a Hybrid App Package* on page 372
- *Hybrid App Packages* on page 367

Transporting Packages Between Environments with Export and Import

If you are trying to transfer deployment packages among disparate server environments, you can export packages from one server environment before importing them into another one. This sequence of events is typically performed by administrators when transporting packages from a development local server environment to a test network environment, or from a test or pilot environment to production cluster environment.

Exporting Hybrid Apps

Export Hybrid Apps to create a deployment archive that can be used to transport Hybrid Apps between SAP Mobile Servers.

1. In the left navigation pane of SAP Control Center, select **Hybrid Apps**.
2. In the right navigation pane, click the **General** tab.
3. Select the box adjacent to the Hybrid App and click **Export**.
4. Click **Next**.
5. Click **Finish**.
6. Select the file system target for the exported contents and click **Save**.

Note: Ensure that you do not hide the file type extension when you name the export archive; otherwise, the *.zip extension becomes invisible, which adversely affects the outcome of the export process.

A status message indicates the success or failure of the export transaction. If the transaction succeeds, a ZIP file is created in the location you specified. You can then import this file on another SAP Mobile Server.

Next

Deliver the file to the appropriate person, or deploy or transport the exported Hybrid App to the appropriate server.

See also

- *Importing Hybrid Apps* on page 370
- *Import Requirements and Best Practices* on page 370

Importing Hybrid Apps

Import Hybrid Apps after they have been exported from another SAP Mobile Server.

1. In the left navigation pane of SAP Control Center, click **Hybrid Apps**.
2. In the right administration pane, click the **General** tab.
3. Click **Import**.
4. Click **Browse** to navigate to the file.
5. Click **Import**.
6. After the import is complete, check the import details and click **OK**.

See also

- *Exporting Hybrid Apps* on page 369
- *Import Requirements and Best Practices* on page 370

Import Requirements and Best Practices

Import is typically used to move a package or application from a development environment to a test environment, and after testing to a production environment.

MBO Package Imports

- **Domain requirements** – all server connections and security configurations referenced by the MBO package must exist in the target domain.
- **Versioning recommendations** – if a developer has updated the package version number:
 1. (Required) Verify that this new package version is added to the application.
 2. (Recommended) Whenever possible, use the update instead of import. Otherwise, delete the existing package first, to remove all runtime data for the package including cached data, registered subscriptions, subscription templates, client log, MBO and operation histories, and registered package users. Delete these items only after serious consideration.

Application Imports

Make sure the target system has resources that match those referenced in the export archive file:

- Domains, security configurations, logical roles assigned to application connection templates, proxy endpoint connections (used by ODP applications)
- If MBO packages are included, the domains, security configurations and connections referenced by the MBO package archives

Hybrid App Imports

If MBO packages are referenced in the export archive file, make sure that the MBO packages have already been deployed to the target system.

Note: If the Hybrid App has matching rules, all matching rule search expressions are imported as regular expression types. Other expression types such as `Begins with` or `Equals` are imported as `Regular expression`.

See also

- *Exporting Hybrid Apps* on page 369
- *Importing Hybrid Apps* on page 370

Enabling and Configuring the Notification Mailbox

Configure the notification mailbox settings that allow SAP Mobile Server to transform e-mail messages into Hybrid App.

The notification mailbox configuration uses a listener to scan all incoming e-mail messages delivered to the particular inbox specified during configuration. When the listener identifies an e-mail message that matches the rules specified by the administrator, it sends the message as a Hybrid App to the device that matches the rule.

Note: Saving changes to the notification mailbox configuration deletes all e-mail messages from the account. Before proceeding with configuration changes, consult your e-mail administrator if you want to back up the existing messages in the configured account.

1. In the left navigation pane, click **Hybrid Apps**.
2. In the right administration pane, click **Notification Mailbox**.
3. Select **Enable**.
4. Configure these properties:
 - **Protocol** – choose between POP3 or IMAP, depending on the e-mail server used.
 - **Use SSL** – encrypt the connection between SAP Mobile Server and the e-mail server in your environment.
 - **Server** and **Port** – configure these connection properties so SAP Mobile Server can connect to the e-mail server in your environment. The defaults are localhost and port 110 (unencrypted) or 995 (encrypted).
 - **User name** and **Password** – configure these login properties so SAP Mobile Server can log in with a valid e-mail user identity.
 - **Truncation limit** – specify the maximum number of characters taken from the body text of the original e-mail message, and downloaded to the client during synchronization. If the body exceeds this number of characters, the listener truncates the body text to the number of specified characters before distributing it. The default is 5000 characters.
 - **Poll seconds** – the number of seconds the listener sleeps between polls. During each poll, the listener checks the master inbox for new e-mail messages to process. The default is 60 seconds.

5. Add at least one distribution rule. You can click **Test** to test your configuration. If the test is successful, click **Save**.

See also

- *Hybrid App Packages* on page 367

Configuring a Hybrid App Package

Configure Hybrid App properties for you production environment.

Prerequisites

You must deploy a package before you can configure properties for it.

1. *Configuring General Hybrid App Properties*

Configure general properties for a Hybrid App, including display name and icon. Alter these settings to change development environment values to production environment equivalents.

2. *Configuring Matching Rules*

Define the parameters and matching rules that determine whether an e-mail message is a regular e-mail, or a Hybrid App e-mail at runtime.

3. *Configuring Context Variables for Hybrid App Packages*

The administrator can change some of the values of a selected variable, should the design-time value need to change for a production environment.

4. *Assigning and Unassigning a Hybrid App to an Application Connection*

Assign a Hybrid App package to an application connection to make it available to a device user. Unassign the Hybrid App package when it is no longer required.

See also

- *Deploying a Hybrid App Package with the Deploy Wizard* on page 368
- *Hybrid App Packages* on page 367

Configuring General Hybrid App Properties

Configure general properties for a Hybrid App, including display name and icon. Alter these settings to change development environment values to production environment equivalents.

1. In the left navigation pane, expand the Hybrid Apps folder and select the Hybrid App to configure the properties for.
2. In the right administration tab, click **General**.

Only an administrator can change the general properties. All others are configured by the Hybrid App developer and cannot be modified.

- **Display name** – sets the name that appears for the Hybrid App package.

- **Display icon** – select from the list of built-in icons. Any custom icons contained in the Hybrid App package can be viewed but not selected.

Note: The Hybrid App package may contain custom icons to accommodate the different image resolutions required by different devices. The use of custom icons is determined by individual devices. The built-in icon is only used if the device cannot display any of the custom icons; for example, if the device does not support the icon image type.

3. Click **Lock/Unlock** to lock or unlock a Hybrid App.

You cannot modify or delete a locked Hybrid App.

4. Click **Save**.

Configuring Matching Rules

Define the parameters and matching rules that determine whether an e-mail message is a regular e-mail, or a Hybrid App e-mail at runtime.

Prerequisites

The developer must have created an object query and added an E-mail Subscription starting point when the Hybrid App was designed.

Task

When a multiplexer, which a user configures for a Notification mailbox, retrieves e-mails from the e-mail server, the "Matching rules" are used to determine if an e-mail is a regular e-mail or a Hybrid App e-mail. If the "matching rules" match, then the e-mail is processed as a Hybrid App e-mail. The e-mail is processed for "Extraction rules" (this processing is not visible in SAP Control Center) to extract values from the e-mail. This determines further processing, such as calling MBO object queries, and so forth. Then, a Hybrid App message is constructed with the necessary data of the object query result, and sent to device(s) according to the "Distribution rules" (which determine the devices to which the message should be sent).

You can configure a matching rule at one of two levels:

- At the inbox level to route e-mails for all Hybrid Apps
 - At the package level to route e-mails only for a specific Hybrid App
1. In the left navigation pane, click **Hybrid Apps** > **Application Name**.
 2. In the right administration pane, click the **Matching Rules** tab.
 3. Configure matching rules by either:
 - Clicking **New** to create a new rule, or,
 - Selecting an existing rule name and clicking **Properties**.
 4. In the **Matching Rules** dialog:

- a) Select the field in the e-mail from which the parameter value is extracted. For example, if you choose From, the parameter value is extracted from the line of the e-mail message that indicates the name of the sender of the message.

You can also select one of the custom parameter values. When registering Hybrid App devices, an administrator can choose one of four device settings. The customer then populates the settings with whatever values they like. Custom parameters can also be set programmatically through Web services.

Note: If you are editing properties of a rule created by a developer, you cannot modify the matching rules.

- b) Choose the type of search expression:
- Equals – the field must exactly match the text in the label.
 - Begins with – the field must begin with the text in the label.
 - Ends with – the field must end with the text in the label.
 - Contains – the text in the label must exist somewhere in the field.
 - Regular expression – search for text that matches the pattern defined by the regular expression. You can create an expression with Boolean operators, groups, or wildcards like "?" or "*". SAP Mobile Platform uses the Boost regular expression engine. See the Boost documentation on regular expression syntax at http://www.boost.org/doc/libs/1_40_0/libs/regex/doc/html/boost_regex/syntax.html.
- c) Configure the text to search against, or define a regular expression in the **Value** field.

Next

Test all new and changed rules to ensure they work as designed.

See also

- *Configuring Context Variables for Hybrid App Packages* on page 375

Testing Configured Matching Rules

Test a new or modified matching rule to ensure it is configured correctly.

1. In the **Matching Rules** tab for a selected Hybrid App package, click **Test**.
2. Populate the fields to create a sample e-mail message against which the rule configuration is tested. Review the results:
 - If a pattern for a corresponding field in the rule matches, the word `Pass` appears adjacent to the field.
 - If a pattern for a corresponding field in the rule does not match, the word `Fail` appears adjacent to the field.
 - Otherwise, `No Rule` appears, indicating that no rule was created for this corresponding e-mail field.

Configuring Context Variables for Hybrid App Packages

The administrator can change some of the values of a selected variable, should the design-time value need to change for a production environment.

Which values are configurable depends on whether the developer hard-coded a set of user credentials or used a certificate.

1. In the left navigation pane, expand the **Hybrid Apps** folder and select the Hybrid App package to configure context variables for.
2. In the right administration pane, click the **Context Variables** tab.
3. Select the context variable to configure, then click **Modify**.

Context Variable	Description
SupUser	The valid Hybrid App application user ID that SAP Mobile Server uses to authenticate the user. Depending on the security configuration, SAP Mobile Server may pass that authentication to an EIS.
SupDomain	The name of the domain that the Hybrid App package is deployed to.
SupUnrecoverableErrorRetryTimeout	After sending a JSON request to SAP Mobile Server, if you receive an EIS code that indicates an unrecoverable error in the response log, the Hybrid App client throws an exception. A retry attempt is made after a retry time interval, which is set to three days by default. Select this property to change the retry time interval.
SupThrowCredentialRequestOn401Error	The default is true , which means that an error code 401 throws a <code>CredentialRequestException</code> , which sends a credential request notification to the user's inbox. If this property is set to false , error code 401 is treated as a normal recoverable exception.
SupThrowBadHttpHeadersOn412Error	The default is true , which means that an error code 412 throws a <code>BadHttpRequestException</code> . If this property is set to false , error code 412 is treated as a normal recoverable exception.

Context Variable	Description
SupRecoverableErrorRetryTimeout	After sending a JSON request to SAP Mobile Server, if you receive an EIS code that indicates a recoverable error in the response log, the Hybrid App client throws an exception. A retry attempt is made after a retry time interval, which is set to 15 minutes by default. Select this property to change the retry time interval.
SupPassword	The valid Hybrid App application user password that SAP Mobile Server uses to authenticate the user. Depending on the security configuration, SAP Mobile Server may pass that authentication to an EIS. An administrator must change development/test values to those required for a production environment.
SupPackages	The name and version of the MBO packages that are used in the Hybrid App. This cannot be changed.
SupMaximumMessageLength	Use this property to increase the allowed maximum Hybrid App message size. Limitations vary depending on device platform: • For BlackBerry 5, the limit is 512KB. • For Windows Mobile the limit is 500KB. • For BlackBerry 6 and Android, the limit depends on the memory condition of the device. Large message may result in an out of memory error.
SupWorkflowVersion	The version number of the Hybrid App package.

4. In the Context Variable dialog, change the value of the named variable and click **OK**.

See also

- *Configuring Matching Rules* on page 373
- *Assigning and Unassigning a Hybrid App to an Application Connection* on page 377

Changing Hard Coded User Credentials

The administrator can change hard coded user credentials assigned at design time if the design time value needs to change for a production environment.

1. In the left navigation pane, expand the **Hybrid Apps** folder and select the Hybrid App package to configure context variables for.

2. In the right administration pane, click the **Context Variables** tab.
3. Select one or both of the variables: SupUser or SupPassword, and click **Modify**.
4. Type the new value and click **OK**.

Adding a Certificate File to the Hybrid App Package

The administrator can change the credential certificate assigned at design time.

Note: SAP recommends that you use Internet Explorer to perform this procedure.

1. In the left navigation pane, expand the **Hybrid Apps** folder and select the Hybrid App package to configure context variables for.
2. In the right administration pane, click the **Context Variables** tab.
3. Select **SupPassword** and click **Modify**.
4. Select **Use certificate-base credentials** and click **Browse** to find and upload a certificate file.
5. Set the value for **Certificate password** and click **OK**.
On the Context Variables page, the read-only values of SupUser, SupCertificateSubject, SupCertificateNotBefore, SupCertificateNotAfter, and SupCertificateIssuer change to reflect values of the new certificate and password you set.

Assigning and Unassigning a Hybrid App to Application Connection Templates

When you assign a Hybrid Apppackage to an application connection template, all application connections that use that template are automatically assigned the Hybrid App package.

1. In the left navigation pane, expand **Hybrid Apps** and select a Hybrid App.
2. Click the **Application Connection Templates** tab.
All application connection templates that the Hybrid App package is assigned to are listed.
3. Click **Assign**.
All available application connection templates are listed.
4. Select one or more application connection templates and click **OK**.

Note: The selected templates must have an application ID.

5. To unassign a Hybrid App from an application connection template, select the template in the **Application Connection Templates** tab and click **Unassign**.

Assigning and Unassigning a Hybrid App to an Application Connection

Assign a Hybrid App package to an application connection to make it available to a device user. Unassign the Hybrid App package when it is no longer required.

You can also assign Hybrid App packages indirectly through the application connection template. The set of packages assigned to an application connection will be a combination of

packages assigned indirectly through the application connection template and directly through the application connection.

1. In the left navigation pane of SAP Control Center, click **Hybrid Apps** and select the Hybrid App to assign.
2. In the right administration pane, click the **Application Connections** tab.
3. Click **Assign**.
4. List the activation users to assign the Hybrid App package to.
Search for users by selecting the user property you want to search on, then selecting the string to match against. Click **Go** to display the users.
5. Select the user or users from the list that you want to assign the Hybrid App package to.
6. Click **OK**.
7. To set the Hybrid App package as the default application for an application connection, select the connection and click **Set Default**.
Set a Hybrid App package as the default to run that application on the device as a single-purpose application. Single-purpose applications launch automatically when the user opens the Hybrid Web Container. This will be the only Hybrid App available on the device. You can select only one default per application connection.
8. To unassign a Hybrid App package, select the application connection and click **Unassign**.

Note: If you unassign the Hybrid App package that is set as the default, you may want to select a new default package.

9. Click **OK**.

See also

- *Configuring Context Variables for Hybrid App Packages* on page 375

Setting and Unsetting a Default Hybrid App

Set a Hybrid App package as the default application for an application connection to run it on the device as a single-purpose application. A single-purpose application launches automatically when the user opens the Hybrid Web Container and is the only Hybrid App available on the device.

1. In the left navigation pane of SAP Control Center, click **Applications**.
2. In the right administration pane, click the **Application Connections** tab.
3. Select the application connection to set the default for.
4. Click **Hybrid Apps**.
5. Select the Hybrid App package to set as the default and click **Set default**.
You can select only one default per application connection.

To remove a Hybrid App package as the default, select the package and click **Unset default**.

6. Click **OK**.

Checking Hybrid App Users and Queues

Check Hybrid App application users and review pending activities for a Hybrid App application.

1. In the left navigation pane, expand the **Hybrid Apps** folder and select the Hybrid App you want to administer.
2. To check Hybrid App users:
 - a) In the right administration pane, select the **Application Connections** tab.
 - b) Review data about Hybrid App device users:
 - User – the name of the user that activates the device. The user under the domain logs should be in the format "username@securityConfiguration"..
 - Application Connection ID – the unique identifier for a user application connection.
 - Errors – the total number of errors on the device.
 - Transform Items – the total number of items in the transform queue. The transform queue contains items that SAP Mobile Server has transformed from e-mail messages into Hybrid App messages to be sent to clients.
 - Response Items – the total number of items in the response queue. The response queue contains Hybrid App messages that are sent from the device to SAP Mobile Server.
 - Transform Queue Status – the current status of the transform queue: active, awaiting credentials, or awaiting retry.
 - Response Queue Status – the current status of the response queue: active, awaiting credentials, or awaiting retry.
3. To view pending activities for a Hybrid App:
 - a) Select the **Queue Items** tab.
 - b) Review data about pending Hybrid App activities:
 - User – the name of the user that activates the device. The user under the domain logs should be in the format "username@securityConfiguration"..
 - Queue Type – the type of Hybrid App queue: response or transform.
 - Application Connection ID – the unique identifier for a user application connection.
 - Logical ID – the unique identifier for a registered mobile device that is generated and maintained by SAP Mobile Server.
 - Queue ID – the unique identifier of the queued item.

- State – the status of the Hybrid App queue: active, awaiting credentials, or awaiting retry.
 - Creation Date – the date the queue item was created.
 - Retry Date – the date that the processing of the queue item is scheduled to be retried (if applicable).
4. To manage the Hybrid App queue in the event of non-recoverable errors:
- a) Select the **Queue Items** tab.
 - b) Identify a Hybrid App queue item that requires you to unblock it or delete it.

Errors affecting Hybrid App queue items are either recoverable (where a retry is applicable) or unrecoverable/unknown (where no automatic retry occurs, or there is very long retry interval).

To recover from a long retry interval, an administrator can unblock a queue currently in retry state, so the next work schedule can pick up the blocked item immediately, instead of waiting for the retry timeout.
 - c) Select one or more of the queue items from the same queue type (the queue type for all the selected items must be either Transform or Response).
 - d) Select one of the following actions:
 - **Delete** – deletes the selected Hybrid App queue item(s).
 - **Unblock** – unblocks the selected Hybrid App queue item(s) that are currently in a retry state.
 - e) Click **OK** to confirm the action.

Agentry Applications

Agentry applications are created by developers using Agentry Editor, and deployed and maintained on an SAP Mobile Server by administrators.

Agentry applications can be deployed:

- When creating the application definition in SAP Control Center
- To an existing application in SAP Control Center (either the initial deployment or an update)

When deploying Agentry applications, you can select:

- **.agp files** – Agentry application definition files.
- **.agpz files** – zip files containing Agentry application definition files and additional development artifacts.
- **.zip files** – files that have been created from existing Agentry application definitions in SAP Control Center by downloading specific Agentry metadata.

See also

- *MBO Packages* on page 339
- *DOE-C Packages* on page 364
- *Hybrid App Packages* on page 367

Creating Agentry Application Definitions

Create an Agentry application definition by providing general application properties such as the application ID, description, security configuration and domain details. You can also deploy the application file to the SAP Mobile Server.

1. In the left navigation pane of SAP Control Center, click the **Applications** node and select the **Applications** tab in the right administration pane.
2. Click **New**.
3. Enter a unique Application ID.
 - SAP recommends that application IDs contain a minimum of two dots ("."). For example, the following ID is valid: `com.sap.mobile.appl`.
 - Application IDs cannot start with a dot ("."). For example, the following ID is invalid: `.com.sap.mobile.appl`.
 - Application IDs cannot have two consecutive dots ("."). For example, the following ID is invalid: `com..sap.mobile.appl`.
4. Select **Agentry** as the Client SDK type.
5. The Security configuration and Domain are preselected and cannot be changed.
The Agentry security configuration is a predefined security configuration assigned to the "default" domain. It uses the `NoSecLoginModule`. This provider forwards login requests to the Agentry Server instance for authentication.
6. (Optional) Select **Application file** and browse to select the file. You can select:
 - **.agp files** – Agentry application definition files.
 - **.agpz files** – zip files containing Agentry application definition files and additional development artifacts.
 - **.zip files** – files that have been created from existing Agentry application definitions in SAP Control Center by downloading specific Agentry metadata.
7. If you are deploying the application file as part of application creation, specify the Active time for the application.
You can activate the application immediately or at a specific date and time in the future. Selecting **Immediate** makes the application available to end users immediately, as long as the application is enabled.
8. Specify whether to **Enable after deployment**.
Selecting this option automatically starts the Agentry Server instance for this application as soon as the application file is deployed (the Agentry Server instance status is `RUNNING`). If the application file has not been deployed, setting this option starts the

Deploy

Agentry Server instance without the application information (the Agentry Server instance status is `RUNNING WITHOUT METADATA`). If you do not select this option, you must manually enable the application before you can start the Agentry Server instance

9. Click **OK** to complete the application definition. Depending on the application definition:
 - The application file is deployed.
 - The Agentry Server instance is started.
 - The application is available for end users.

Note: Any deploy errors or warnings are recorded in the SAP Mobile Server log. This log is located in `SMP_HOME\Servers\UnwiredServer\logs\<hostname>-server.log` or can be viewed in SAP Control Center under **Servers > primaryServer > Log**.

Deploying Agentry Application Files to an Existing Application

If you created an Agentry application without deploying the application file, or to deploy an updated application file, you can deploy to an existing application.

1. In the navigation pane of SAP Control Center, expand the **Applications** node and select the Agentry application that you want to deploy to.
2. In the administration pane, click the **General** tab.
3. Click **Deploy**.
4. Browse to select the application file. You can select:
 - **.agp files** – Agentry application definition files.
 - **.agpz files** – zip files containing Agentry application definition files and additional development artifacts.
 - **.zip files** – files that have been created from existing Agentry application definitions in SAP Control Center by downloading specific Agentry metadata.
5. Specify the Active time for the application.

You can activate the application immediately or at a specific date and time in the future. Selecting **Immediate** makes the application available to end users immediately, as long as the application is enabled.

6. Click **OK** to deploy the application file.

If this is the initial deployment and the application is enabled, the Agentry Server instance status changes to `RUNNING`.

If this is an update to a running application, the new or changed file is created or updated and can be managed in SAP Control Center.

Note: Any deploy errors or warnings are recorded in the SAP Mobile Server log. This log is located in `SMP_HOME\Servers\UnwiredServer\logs\<hostname>-`

`server.log` or can be viewed in SAP Control Center under **Servers** > **primaryServer** > **Log**.

Next

After deployment, you can review the application metadata that was downloaded by clicking the **Download** button on the **General** tab of the application.

Downloading Agentry Application Metadata for Deploying

Once you have create an Agentry application definition in SAP Control Center, you can download specific Agentry application metadata to include in a .zip file. The .zip file can then be deployed in another server environment.

Download instead of exporting when you need more control over the application metadata that is being packaged for deployment, for example, when providing updates of specific files for an application in production.

1. In the navigation pane of SAP Control Center, expand the **Applications** node and select the application you want to create a .zip file for.
2. In the administration pane, click the **General** tab.
3. Click **Download**.
4. Select the application definition files, locales, and third-party resources to package for deployment.

Note: You must make at least one selection.

5. Click **Next**, then **Finish**.
6. Select the location for the zip file and change the name as required.
7. Click **Save**.

You can now deploy the zip file in another environment. You must define the application in the target environment before deploying the file.

See also

- *Creating Agentry Application Definitions* on page 267

Transporting Applications Between Environments Using Export and Import

Transport applications between different server environments by exporting them from one server, before importing them into another one. This sequence of events is typically performed by administrators when transporting applications from a development local server environment to a test network environment, or from a test or pilot environment to a production cluster environment.

Exporting Applications

Export applications to create a deployment archive that can be used to transport applications between SAP Mobile Servers.

Prerequisites

Before beginning, review import requirements and best practices.

Task

1. In the left navigation pane of SAP Control Center, select **Applications**.
2. In the right navigation pane, click the **Applications** tab.
3. Select the box adjacent to the application and click **Export**.
4. Click **Next**.
5. Click **Finish**.
6. Select the file system target for the exported contents and click **Save**.

Note: Ensure that you do not hide the file type extension when you name the export archive; otherwise, the *.zip extension becomes invisible, which adversely affects the outcome of the export process.

A status message indicates the success or failure of the export transaction. If the transaction succeeds, a ZIP file is created in the location you specified. You can then import this file on another SAP Mobile Server.

Next

Deliver the file to the appropriate person, or deploy or transport the exported application to the appropriate server.

Importing Applications

Import applications after they have been exported from another SAP Mobile Server.

Prerequisites

Review import requirements and best practices before beginning.

Task

Note: Only platform administrators can import applications at the cluster level.

1. In the navigation pane of SAP Control Center, select the **Applications** node.
2. In the right administration pane, click the **Applications** tab and click **Import**.
3. Click **Browse** to navigate to the file.

4. Click **Import**.
5. After the import is complete, check the import details and click **OK**.

Next

If the application connection template does not exist on the target server, it is imported. You may need to change the connection and proxy settings to match the target server configuration. If the application connection template already exists on the target server, connection and proxy settings are not imported; the target server template settings are used for connection and proxy settings.

Deploy

Monitor

Monitor availability status, view system and performance statistics, and review system data that allow administrators to diagnose the health and security of the runtime.

Monitored operations include security, replication-based synchronization, messaging-based synchronization, messaging queue, data change notification, device notification, package, user, and cache activity. These aspects of monitoring are important to ensuring that the required data is collected.

The critical aspects of monitoring include:

1. Setting up a monitoring configuration. A monitoring configuration sets the server behavior for writing data to database, automatic purge, and data source where the monitoring data is stored.

A default configuration is created for you, however you will likely want to customize this configuration for your environment. By default, monitoring data is flushed every 5 minutes. In development and debugging scenarios, you may need to set the flush behavior to be immediate. Set the **Number of rows** and **Batch size** properties to a low number. You can also disable flush, which results in immediately persisting changes to monitoring database. If you are setting up immediate persistence in a production environment, you may experience degraded performance. Use persistence with caution.

2. Creating a monitoring profile. A monitoring profile defines one or more domains and packages that need to be monitored.

You can either use the **default** profile to capture monitoring data for all packages in all domains or create specific profiles as required. Otherwise, disable the **default** profile or modify it as needed.

3. Reviewing the captured data. An administrator can review monitoring data (current, historical, and performance statistics) from SAP Control Center.

Use the monitoring tabs to filter the data by domain, package, and time range. You can also export the data into a CSV or XML file and then use any available reporting or spreadsheet tool to analyze the data.

Monitoring Usage

Monitoring information reflects current and historical activity, and general performance during a specified time period.

Monitoring allows administrators to identify key areas of weakness or periods of high activity in the particular area they are monitoring. Access to this data helps administrators make decisions about how to better configure the application environment to achieve a higher level of performance.

The historical data is preserved in the monitor database. Performance data (KPIs for Replication, Messaging, Package Statistics, User Statistics, and Cache Statistics) for the specified time period is calculated upon request using the historical data available for that period. If monitoring data is purged for that time period, the performance data calculations will not factor in that data. It is recommended to purge monitoring data after putting in place mechanisms to export the required historical and/or performance data as needed. By default, monitoring data is automatically purged after seven days.

Also note that the processing times are calculated based on the time the request (or message) arrives on the server, and the time it took to process the request (or message) on the server. The client-side time (request origin time, and time taken to deliver to the server) are not factored into that data.

See also

- *System Monitoring Overview* on page 388
- *Monitoring Configuration* on page 390
- *Monitoring Profiles* on page 392
- *Monitoring Data* on page 394

System Monitoring Overview

(Not applicable to Online Data Proxy) The goal of monitoring is to provide a record of activities and performance statistics for various elements of the application. Monitoring is an ongoing administration task.

Use monitoring information to identify errors in the system and resolve them appropriately. This data can also be shared by platform and domain administrators by exporting and saving the data to a .CSV or .XML file.

Note: Monitoring is only supported for object API applications. The equivalent support for OData applications is in domain logging. There are specific log buckets for OData based applications (Proxy Request/Response, Push, Error). To set up domain logging for OData, you configure a domain log in the default domain, create a domain log profile in a domain where application exists to specify the logging scope, and use SAP Control Center to trace the activities. See *SAP Control Center for SAP Mobile Platform > Administer > Domains > Domain Logs*.

The platform administrator uses SAP Control Center to monitor various aspects of SAP Mobile Platform. Monitoring information includes current activity, historical activity, and general performance during a specified time period. You can monitor these components:

- Security log
- Replication synchronization
- Messaging synchronization

- System messaging queue status
- Data change notifications
- Device notifications (replication)
- Package statistics (replication and messaging)
- User-related activity
- Cache activity

To enable monitoring, platform administrators must set up a monitoring database, configure a monitoring data source or create a new one, and set up monitoring database flush and purge options. By default the installer created a monitoring database, however you can use another one if you choose. You must activate an application with a registered connection before you can monitor it.

To control monitoring, platform administrators create monitoring profiles and configurations, which define the targets (domains and packages) to monitor for a configured length of time. A default monitoring profile is created for you by the installer. Monitoring data can be deleted by the platform administrator as needed.

Table 23. System monitoring tasks

Task	Frequency	Accomplished by
Create and enable monitoring profiles	One-time initial configuration with infrequent tuning as required	SAP Control Center for SAP Mobile Platform with the Monitoring node
Enable domain logging	One-time setup with infrequent configuration changes, usually as issues arise	SAP Control Center for SAP Mobile Platform with the Domain node. Expand the node and select <domainName> > Log .
Review current/historical/performance metrics	Routine	SAP Control Center for SAP Mobile Platform with the Monitoring node
Identify performance issues	Active	SAP Control Center for SAP Mobile Platform with the Monitoring node
Monitor application and user activity to check for irregularities	Active	SAP Control Center for SAP Mobile Platform with the Monitoring node
Troubleshoot irregularities	Infrequent	Reviewing various platform logs
Purge or export data	On demand	SAP Control Center for SAP Mobile Platform with the Monitoring node

See also

- *Monitoring Usage* on page 387
- *Monitoring Configuration* on page 390
- *Monitoring Profiles* on page 392
- *Monitoring Data* on page 394

Monitoring Configuration

The monitoring configuration identifies the monitoring database endpoint and determines how long data is stored in the database.

The configurable monitoring properties are:

- Auto-purge – configures an automatic purge of the monitoring database to occur on a regular basis.
- Flush threshold – determines how often monitoring data is flushed from the server memory for storage in the monitoring database.
- Flush batch size – divides flushed data into smaller segments, rather than saving all data together according to the flush threshold parameters.
- Monitor database endpoint – sets the database to be used for storage of monitoring data.

See also

- *Monitoring Usage* on page 387
- *System Monitoring Overview* on page 388
- *Monitoring Profiles* on page 392
- *Monitoring Data* on page 394

Configuring Monitoring Performance Properties

Configure auto-purge, flush threshold, and flush batch size settings to determine how long monitoring data is retained, and set a monitoring database to configure where data is stored.

Prerequisites

Depending on the expected level of monitoring activity, ensure that the monitoring database is adequately prepared to store monitoring data.

Task

1. In the left navigation pane of SAP Control Center, select **Monitoring**.
2. In the right administration pane, select the **General** tab.
3. Click **Configuration**.

4. Configure auto purge settings.

Auto purge clears obsolete data from the monitoring database once it reaches the specified threshold.

- a) Select **Enable auto purge configuration** to activate auto purge functionality.
- b) Enter the length of time (in days) to retain monitoring data before it is purged.

5. Configure flush threshold settings.

The flush threshold indicates how often data is flushed from memory to the database. This allows you to specify the size of the data saved in memory before it is cleared. Alternately, if you do not enable a flush threshold, data is automatically written to the monitoring database as it is captured.

- a) Select **Enable flush threshold configuration** to activate flush threshold functionality.
- b) Select one of:

- **Number of rows** – monitoring data that surpasses the specified number of rows is flushed from memory. Enter the desired number of rows adjacent to **Rows**. The default is 100.
- **Time interval** – monitoring data older than the specified time interval is flushed from memory. Enter the desired duration adjacent to **Minutes**. The default is 5.
- **Either rows or time interval** – monitoring data is flushed from memory according to whichever value is reached first: either the specified number of rows or the specified time interval. Enter the desired rows and duration adjacent to **Rows** and **Minutes**, respectively.

6. If you enabled a flush threshold, enter a **Flush batch row size** by specifying the size of each batch of data sent to the monitoring database. The row size must be a positive integer.

The batch size divides flushed data into smaller segments, rather than saving all data together according to the flush threshold parameters. For example, if you set the flush threshold to 100 rows and the flush batch row size to 50, once 100 rows are collected in the monitoring console, the save process executes twice; data is flushed into the database in two batches of 50 rows. If the flush threshold is not enabled, the flush batch row size is implicitly 1.

Note: By default, the monitoring database flushes data every 5 minutes. Alternatively, you can flush data immediately by removing or decreasing the default values, but doing so impacts performance and prevents you from using captured data.

7. Optional. To change the data source, select an available database from the **Monitor database endpoint** drop down list.

Available databases are those with a JDBC server connection type created in the "default" domain. To create a new monitor database, a platform administrator must set up a database by running the appropriate configuration scripts and creating a server connection for the database in the default domain. The database server connection then appears as an option in the Monitor Database Endpoint drop down list.

8. Click **OK**.

Monitoring Profiles

Monitoring profiles specify a monitoring schedule for a particular group of packages. These profiles let administrators collect granular data on which to base domain maintenance and configuration decisions.

A default monitoring profile is automatically created in disabled state on SAP Mobile Server. Administrators can enable or remove the default profile, and enable one or more new monitoring profiles as required.

The same monitoring schedule can be applied to packages across different domains; similarly, you can select individual packages for a monitoring profile.

See also

- *Monitoring Usage* on page 387
- *System Monitoring Overview* on page 388
- *Monitoring Configuration* on page 390
- *Monitoring Data* on page 394

Creating and Enabling a Monitoring Profile

Specify a monitoring schedule for a group of packages.

Prerequisites

Depending on the expected level of monitoring activity, ensure that the monitoring database is adequately prepared to store monitoring data.

Task

1. In the left navigation pane, select **Monitoring**.
2. In the right administration pane, select the **General** tab.
3. Click **New** to create a monitoring profile.
4. Enter a name for the new profile.
5. Select the **Domains and Packages** tab and choose packages to be monitored according to these options:
 - Monitor all domains and packages – select **All Domains and Packages**.
 - Monitor all packages from one or more domains – select a domain, then click **Select All Packages**. Perform this step for each domain you want to monitor.
 - Monitor specific packages from one or more domains – select a domain, then select the particular packages you want to monitor from that domain. Perform this step for each domain you want to monitor.

6. Select **View my selections** to view the packages you selected for the monitoring profile. Unselect this option to return to the package selection table.
7. Select **Enable after creation** to enable monitoring for the selected packages immediately after you create the profile. By default, this option is selected. Unselect this option to enable the monitoring profile later.
8. On the **Schedule** tab, select a schedule to specify when monitoring takes place:
 - **Always On** – this schedule requires no settings. Package activity is continually monitored.
 - **Run Once** – specify a length of time during which monitoring occurs, in either minutes or hours. Package activity is monitored for the duration specified for one time only.
 - **Custom** – specify start and end dates, start and end times, and days of the week. Package activity is monitored according to the time frame specified. See *Setting a Custom Monitoring Schedule*.
9. Click **OK**.
A status message appears in the administration pane indicating the success or failure of profile creation. If successful, the profile appears in the monitoring profiles table.
10. To enable a profile that you did not enable during creation, select the monitoring profile and click **Enable**.

Setting a Custom Monitoring Schedule

Customize the monitoring schedule for packages within a monitoring profile. Setting a custom schedule is the most flexible option; monitoring information is provided according to the time frame you specify.

Prerequisites

Begin creating a monitoring profile in the New Monitor Profile dialog.

Task

1. In the New Monitor Profile dialog, select the **Schedule** tab.
2. Select **Custom** as the monitoring schedule criteria.
3. To set a range to control which days the custom schedule runs, configure a start date and time, end date and time, or day of week (if applicable).
 - Select **Start Date** to set a date for when monitoring of package activity begins. To be more specific, you can also enter a **Start Time**. In this case, monitoring cannot begin until a given time on a given day has been reached.
 - Select **End Date** to set a date that ends the monitoring of package activity. To be more specific, you can also enter an **End Time**.
 - Select the days of the week that package monitoring runs. This means that for the days you select, the schedule runs every week on the day or days you specify.

If you do not indicate a time frame, SAP Mobile Server uses the default custom schedule, which is equivalent to Always On monitoring.

4. Click **OK**.

Monitoring Data

Monitoring data is aggregated in the Monitoring node of SAP Mobile Server and organized by activity, including security, replication-based synchronization, messaging-based synchronization, messaging queue, data change notifications, device notifications, packages, users, and cache. The data for each activity is further broken down into current, historical, and performance-related information. View data for each monitored activity to track the performance and health of the system.

You can selectively view data accrued during a specific time period to see a snapshot of system performance during specific periods. The export function allows you to save data to a file outside of SAP Control Center for reference or logging purposes.

See also

- *Monitoring Usage* on page 387
- *System Monitoring Overview* on page 388
- *Monitoring Configuration* on page 390
- *Monitoring Profiles* on page 392

Reviewing System Monitoring Data

Review data for monitored activities. The monitoring data is retrieved according to the specified time range. Key Performance Indicators (KPIs) are also calculated for the specified time range.

1. In the left navigation pane, select **Monitoring**.
2. In the right administration pane, select one of the following tabs according to the type of monitoring data you want to view:
 - **Security Log**
 - **Replication**
 - **Messaging**
 - **Queue**
 - **Data Change Notifications**
 - **Device Notifications**
 - **Package Statistics**
 - **User Statistics**
 - **Cache Statistics**

See also

- *Purging Monitoring Data* on page 395
- *Exporting Monitoring Data* on page 395
- *Searching Monitoring Data* on page 396
- *Monitoring Data Categories* on page 397

Purging Monitoring Data

Clear old data from the monitoring database.

Using the Purge function in SAP Control Center allows you to perform an ad hoc purge of monitoring data.

1. In the left navigation pane, select **Monitoring**.
2. In the right administration pane, select the **General** tab.
3. Click **Purge**.
4. Indicate the time period for which you want to delete data by specifying a **Start Date**, **Start Time**, **End Date**, and **End Time**.

All monitoring data collected from the start time and date to the end time and date is deleted from the database. To save the data to file before purging it, see *Exporting Monitoring Data*.

5. Click **OK**.

A status message appears in the right administration pane indicating that the data purge was successfully completed.

See also

- *Reviewing System Monitoring Data* on page 394
- *Exporting Monitoring Data* on page 395
- *Searching Monitoring Data* on page 396
- *Monitoring Data Categories* on page 397

Exporting Monitoring Data

Save a segment of monitoring data to a location outside of the monitoring database. Export data to back up information, particularly before purging it from the database, or to perform closer analysis of the data in a spreadsheet application.

This option is especially useful when you need to share monitoring data with other administrators and tenants. Since this task can be time-consuming, depending upon the size of the data being exported, SAP recommends that you export the data in segments or perform the export at a time when SAP Control Center is not in use.

Note: The time taken to export the requested data is dependent on the time range specified, and the amount of data in the monitoring database. If the export is taking too long, or the user

interface is blocked for too long, another option is to export the monitoring data using the management APIs provided for exporting monitoring data. Please see *Developer Guide: SAP Mobile Server Runtime > Management API* for further details.

1. In the left navigation pane, select **Monitoring**.
2. In the right administration pane, select the tab corresponding to the monitoring data you want to view.
3. Perform a search using the appropriate criteria to obtain the desired monitoring data.
4. Click **Export**.
5. Select a file type for the exported data (CSV or XML), and click **Next**.
6. Click **Finish**.
7. In the file browser dialog, select a save location and enter a unique file name.
8. Click **Save**.

All monitoring data retrieved by the search is saved to the file you specify in step 7.

See also

- *Reviewing System Monitoring Data* on page 394
- *Purging Monitoring Data* on page 395
- *Searching Monitoring Data* on page 396
- *Monitoring Data Categories* on page 397

Searching Monitoring Data

Filter monitoring data according to a specified date and time range.

Filter options vary depending upon the type of monitoring data you search.

1. In the left navigation pane, select **Monitoring**.
2. In the right administration pane, select the tab corresponding to the monitoring data you want to view.
3. In the Search pane, indicate the time period for which you want to view data by specifying a date range to search within (that is, Start Date, Start Time, End Date, and End Time) .

Note: You do not need to specify a time period if you are performing a search on Current date.

4. Filter the system components to include in the search.
 - a) Select **Show filter**.
 - b) Specify the components to include in your search.

If a domain or package does not appear in the search list (for example, if it has been deleted), enter the name and click **Add**.

5. From the **Sort By** drop-down list, select a category by which to sort search results.

Note: This field is disabled for some categories.

6. To view the domains and packages you included in the search, select **View my selections**.
7. Click **Retrieve**.

Monitoring data for the time period specified appears in the administration console.

See also

- *Reviewing System Monitoring Data* on page 394
- *Purging Monitoring Data* on page 395
- *Exporting Monitoring Data* on page 395
- *Monitoring Data Categories* on page 397

Viewing Package-Level Cache Statistics

Use the package tree to view cache statistics at the package, cache group, or mobile business object (MBO) level.

The package tree allows for a granular view of data in all cache statistic categories except for domain-level data. Domain-level data instead uses the Filter by Domain search functionality.

1. In the left navigation pane, select **Monitoring**.
2. In the right administration pane, select **Cache Statistics**.
3. From the cache feature drop-down list, select one of the following, depending on the type of data and level of granularity you require:
 - **Domain level**
 - **Package level**
 - **Package level cache group**
 - **Package level MBO**
4. Select **Show filter**.

The tree view appears on the left side of the right administration pane.
5. In the tree view, click the package, cache group, or MBO for which you want to view monitoring data.

The monitoring data displays in the monitoring console. You can further filter data by specifying a time period in the search panel (for package-level cache performance and package-level MBO status only).

Monitoring Data Categories

Monitoring data is organized according to object type, allowing administrators to perform focused data analysis on specific activities and SAP Mobile Platform components. Current,

historical, and performance-based statistics facilitate user support, troubleshooting, and performance tracking for individual application environments.

The replication and messaging categories are the primary sources of data relating to application environment performance. The remaining tabs present detailed monitoring data that focuses on various aspects of replication-based applications, messaging-based applications, or both.

See also

- *Reviewing System Monitoring Data* on page 394
- *Purging Monitoring Data* on page 395
- *Exporting Monitoring Data* on page 395
- *Searching Monitoring Data* on page 396

Security Log Statistics

The security log reflects the authentication history of users either across the cluster, or filtered according to domain, during a specified time period. These statistics allow you to diagnose and troubleshoot connection or authentication problems on a per-user basis. Security log monitoring is always enabled.

User security data falls into these categories:

Category	Description
User	The user name
Security Configuration	The security configuration to which the device user belongs
Time	The time at which the authentication request took place
Result	The outcome of the authentication request: success or failure
Application Connection ID	The application connection ID associated with the user
Package	The package the user was attempting to access
Domain	The domain the user was attempting to access

See also

- *Replication Statistics* on page 399
- *Messaging Statistics* on page 403
- *Messaging Queue Statistics* on page 407
- *Data Change Notification Statistics* on page 408
- *Device Notification Statistics* on page 410
- *Package Statistics* on page 412
- *User Statistics* on page 414

- *Cache Statistics* on page 416

Replication Statistics

Replication statistics reflect replication synchronization activity for monitored packages. Current statistics monitor the progress of real-time synchronizations, while historical statistics present data from completed synchronizations on a per-package basis. Performance monitoring uses key performance indicators to produce data about synchronization efficiency.

Through statistics that report on the duration and scope of synchronizations, as well as any errors experienced during synchronization, replication monitoring allows you to identify the rate at which synchronizations happen during specified time periods, which users synchronize data, and which mobile business objects are affected.

See also

- *Security Log Statistics* on page 398
- *Messaging Statistics* on page 403
- *Messaging Queue Statistics* on page 407
- *Data Change Notification Statistics* on page 408
- *Device Notification Statistics* on page 410
- *Package Statistics* on page 412
- *User Statistics* on page 414
- *Cache Statistics* on page 416

Current Replication Statistics

Current statistics for replication synchronization provide real-time information about in-progress synchronizations.

SAP Mobile Server monitors replication requests using these statistical categories:

Category	Description
Application ID	The ID associated with the application.
Package	The package name.
Phase	The current synchronization activity: upload or download. During the upload phase, a client initiates operation replays to execute mobile business object (MBO) operations on the back-end system. During the download phase, a client synchronizes with SAP Mobile Server to receive the latest changes to an MBO from the back-end system.
Entity	During the download phase, the name of the MBO with which the client is synchronizing. During the upload phase, the name of the operation that the client is performing.

Category	Description
Synchronization Start Time	The date and time that the synchronization request was initiated.
Domain	The domain to which the package involved in synchronization belongs.
Application Connection ID	The ID number of the connection participating in the synchronization.
User	The name of the user associated with the device ID.

Replication History Statistics

Historical data for replication-based synchronization consists of past synchronization details for monitored packages.

The summary view provides general information, whereas the detail view presents a more specific view of all request events during each synchronization; each row of data corresponds to a synchronization request from the client in the time frame you define:

- Click either **Details** to see more granular information on each synchronization request, or select the **Detail view** option to see all synchronization request details. Detail view allows you to look at the individual messages that make up the summary view.
- Select the **Summary view** option to see aggregated details by domain, package, and user about past synchronization events for the defined time frame.

Table 24. Detail view information

Synchronization element	Description
Application ID	The ID number associated with an application.
Package	The package name.
Application Connection ID	The ID number of the connection used in a synchronization request.
User	The user associated with the device ID.
Phase	The sync activity that occurred during this part of synchronization: upload or download. During the upload phase, a client initiates operation replays to change an MBO. During the download phase, a client synchronizes with SAP Mobile Server to receive the latest changes to an MBO.
Entity	During download, the name of the MBO that the client is synchronizing with. During upload, the operation that the client is performing: create, update, or delete.

Synchronization element	Description
Total Rows Sent	The total number of rows sent during package synchronization. This data type is not supported at the MBO level.
Bytes Transferred	The amount of data transferred during the synchronization request.
Start Time	The date and time that the synchronization request was initiated.
Finish Time	The date and time that this part of synchronization completed.
Error	The incidence of errors during this request: true or false.
Domain	The domain to which the package involved in synchronization belongs.

Table 25. Summary view information

Category	Description
Application ID	The ID number associated with an application.
User	The name of the user associated with the device ID.
Package	The package name.
Total Rows Sent	The total number of rows sent during package synchronization.
Total Operation Replays	The total number of operation replays performed by clients during synchronization.
Total Bytes Sent	The total amount of data (in bytes) downloaded by clients from SAP Mobile Server during synchronization.
Total Bytes Received	The total amount of data (in bytes) uploaded to SAP Mobile Server by clients during synchronization.
Start Time	The date and time that the synchronization request was initiated.
Total Synchronization Time	The amount of time taken to complete the synchronization.
Total Errors	The total number of errors that occurred for the package during synchronization.
Domain	The domain to which the package involved in synchronization belongs.

Replication Performance Statistics

Replication performance statistics consist of key performance indicators (KPIs) that reflect the overall functioning of the application environment.

Performance monitoring highlights key totals and identifies average, minimum, and maximum values for primary activities. These calculations are dynamic, and are based on the data currently available in monitoring database for the specified time period.

All values in this table (totals, averages, maximums, minimums) apply to the specific time period you indicate:

KPI	Description
Total Distinct Package Synchronized	The total number of packages subject to synchronization.
Total Distinct Users	The total number of users who initiated synchronization requests. This value comprises only individual users, and does not count multiple synchronizations requested by the same user.
Average/Minimum/Maximum Synchronization Time	The average, minimum, or maximum amount of time SAP Mobile Server took to finish a complete synchronization.
Time at Minimum/Maximum Synchronization Time	The time of day at which the shortest or longest synchronization completed.
Package with Minimum/Maximum Synchronization Time	The name of the package and associated MBO with the shortest or longest synchronization time.
Average/Minimum/Maximum MBO Rows Per Synchronization	The average, minimum, or maximum number of MBO rows of data that are downloaded when synchronization completes.
Average/Minimum/Maximum Operation Replies per Synchronization (records received)	The average, least, or greatest number of operation replays per synchronization received by SAP Mobile Server from a client.
Total Bytes Sent	The total number of bytes downloaded by clients from SAP Mobile Server.
Total Bytes Received	The total number of bytes uploaded from clients to SAP Mobile Server.
Total Operation Replies	The total number of operation replays performed on the EIS.
Total Errors	The total number of errors that took place across all synchronizations.

KPI	Description
Average/Minimum/Maximum Concurrent Users	The average, least, or greatest number of users involved in concurrent synchronizations.
Time at Minimum/Maximum Concurrent Users	The time at which the least or greatest number of users were involved in concurrent synchronizations.

Messaging Statistics

Messaging statistics report on messaging synchronization activity for monitored packages.

- Current monitoring data tracks the progress of messages from device users presently performing operation replays or synchronizing MBOs.
- Historical data reveals statistics indicating the efficiency of completed transactions.
- Performance monitoring provides an overall view of messaging payload activity intended to highlight areas of strength and weakness in the application environment.

Messaging historical data captures messages such as login, subscribe, import, suspend, resume and so on. The Import type message is a data payload message from server to client (outbound messages), while rest of the messages (login, subscribe, replay, suspend, resume) are sent from the client to server (inbound messages).

See also

- *Security Log Statistics* on page 398
- *Replication Statistics* on page 399
- *Messaging Queue Statistics* on page 407
- *Data Change Notification Statistics* on page 408
- *Device Notification Statistics* on page 410
- *Package Statistics* on page 412
- *User Statistics* on page 414
- *Cache Statistics* on page 416

Current Messaging Statistics

Current statistics for messaging synchronization provide real-time information about in-progress synchronizations. Because messaging synchronizations progress rapidly, there is typically little pending messaging data available at any given time.

SAP Mobile Server monitors messaging requests using these categories:

Category	Description
Application ID	The ID associated with the application.
Package	The package name.

Category	Description
Message Type	The type of message sent by the client to SAP Mobile Server, indicating the current sync activity; for example, import, replay, subscribe, suspend, resume, and so on.
Entity	During the import process, the name of the mobile business object (MBO) with which the client is synchronizing. During replay, the operation that the client is performing. For all other message types, the cell is blank.
Start Time	The date and time that the initial message requesting synchronization was sent by the client.
Domain	The domain to which the package involved in synchronization belongs.
Application Connection ID	The ID number of the application participating in the synchronization.
User	The name of the user associated with the device ID.

Messaging History Statistics

Historical data for messaging synchronization consists of past synchronization details for monitored packages.

The summary view provides general information, whereas the detail view presents a more specific view of all request events during each synchronization; each row of data corresponds to a synchronization request from the client in the time frame you define:

- Click either **Details** to see more granular information on each synchronization request, or select the **Detail view** option to see all synchronization request details. Detail view allows you to look at the individual messages that make up the summary view.
- Select the **Summary view** to see aggregated details by domain, package, and user about past synchronization events for the defined time frame.

Table 26. Detail view information

Data type	Description
Application ID	The ID number associated with an application.
Package	The package name.
Application Connection ID	The ID number of the connection that participated in the synchronization request.
User	The name of the user associated with the device ID.

Data type	Description
Message Type	The type of message sent by the client to SAP Mobile Server, indicating the sync activity; for example, import, replay, subscribe, suspend, resume, and so on.
Entity	During the import process, the name of the mobile business object (MBO) that the client is synchronizing with. During replay, the operation that the client is performing. For all other message types, the cell is blank.
Payload Size	The size of the message (in bytes).
Start Time	The date and time that the message for this sync request is received.
Finish Time	The date and time that the message for this sync request is processed.
Processing Time	The total amount of time between the start time and the finish time.
Error	The incidence of errors during this request; either true or false.
Domain	The domain to which the package involved in synchronization belongs.

Table 27. Summary view information

Category	Description
Application ID	The ID number associated with an application.
User	The name of the user associated with the device ID
Package	The package name
Total Messages Sent	The total number of messages sent by SAP Mobile Server to clients during synchronization
Total Messages Received	The total number of messages received by SAP Mobile Server from clients during synchronization
Total Payload Size Sent	The total amount of data (in bytes) downloaded by clients from SAP Mobile Server during synchronization
Total Payload Size Received	The total amount of data (in bytes) uploaded to SAP Mobile Server by clients during synchronization
Total Operation Replays	The total number of operation replays performed by clients during synchronization
Last Time In	The date and time that the last inbound request was received

Category	Description
Last Time Out	The date and time that the last outbound response was sent
Subscription Commands Count	The total number of subscription commands sent during synchronization; for example, subscribe, recover, suspend, and so on
Total Errors	The number of errors that occurred for the package during synchronization
Domain	The domain to which the package involved in synchronization belongs

Messaging Performance Statistics

Messaging performance statistics consist of key performance indicators (KPIs) that reflect the overall functioning of the application environment.

Performance monitoring highlights key totals and identifies average, minimum, and maximum values for primary activities. These calculations are dynamic, and are based on the data currently available in monitoring database for the specified time period.

All values in this table (totals, averages, maximums, minimums) apply to the specific time period you indicate:

KPI	Description
Total Messages	The total number of messages sent between the server and clients during synchronization.
Total Distinct Devices	The total number of devices involved in synchronization. This total includes the same user multiple times if he or she has multiple devices. The value comprises individual devices, and does not count multiple synchronizations requested by the same device.
Total Distinct Users	The total number of users who initiated synchronization requests. This value comprises individual users, and does not count multiple synchronizations requested by the same user if he or she uses multiple devices.
Average/Minimum/Maximum Concurrent Users	The average, minimum, or maximum number of users involved in simultaneous synchronizations.
Time at Minimum/Maximum Concurrent Users	The time at which the greatest or least number of users were involved in concurrent synchronizations.
Average/Minimum/Maximum Processing Time	The average, minimum, or maximum amount of time SAP Mobile Server took to respond to a sync request message.

KPI	Description
Time at Minimum/Maximum Message Processing Time	The time of day at which the shortest or longest message processing event completed.
MBO for Maximum/Minimum Message Processing Time	The name of the package and associated mobile business object (MBO) with the shortest or longest message processing time.
Average/Minimum/Maximum Message Size	The average, smallest, or largest message sent during synchronization.
Total Inbound Messages	The total number of messages sent from clients to SAP Mobile Server.
Total Outbound Messages	The total number of messages sent from SAP Mobile Server to clients.
Total Operation Replays	The total number of operation replays performed by clients on MBOs.
Total Errors	The total number of errors that took place across all synchronizations.

Note: Reporting of KPIs related to concurrent users is based on a background task that takes a periodic snapshot of the messaging activities. Depending on the nature and length of the processing of a request, the background snapshot may not always see all the requests.

Messaging Queue Statistics

Messaging queue statistics reflect the status of various messaging queues. The data does not reveal any application-specific information, but provides a historical view of messaging activities that communicates the efficiency of messaging-based synchronization, as well as the demands of device client users on the system.

Based on this data, administrators can calculate the appropriate inbound and outbound message queue counts for the system (configurable in the Server Configuration node of SAP Control Center).

See also

- *Security Log Statistics* on page 398
- *Replication Statistics* on page 399
- *Messaging Statistics* on page 403
- *Data Change Notification Statistics* on page 408
- *Device Notification Statistics* on page 410
- *Package Statistics* on page 412
- *User Statistics* on page 414
- *Cache Statistics* on page 416

Messaging Queue Status

Messaging queue status data provides historical information about the processing of messaging-based synchronization requests by SAP Mobile Server. The data indicates areas of high load and times of greatest activity. This data can help administrators decide how to handle queue congestion and other performance issues.

These key indicators monitor messaging queue status:

Statistic	Description
Name	The name of the messaging queue.
Current Queued Items	The total number of pending messages waiting to be processed by SAP Mobile Server.
Average/Minimum/Maximum Queue Depth	The average, minimum, or maximum number of queued messages. For minimum and maximum queue depth, this value is calculated from the last server restart.
Time at Minimum/Maximum Queue Depth	The time and date at which the queue reached its minimum or maximum depth.
Type	The direction of message flow: inbound or outbound.
Total Messages	The total number of messages in the queue at one point since the last server reboot.
Bytes Received	The total number of bytes processed by the queue since the last server reboot.
Last Activity Time	The time at which the most recent message was added to the queue since the last server reboot.

Data Change Notification Statistics

Data change notification (DCN) statistics monitor notifications that are received by SAP Mobile Server from the enterprise information server. Specifically, DCN monitoring reports which packages and sync groups are affected by notifications, and how quickly these are processed by the server.

Monitoring DCN statistics allows you to troubleshoot and diagnose performance issues if, for example, the cache is not being updated quickly enough. These statistics help to identify which packages took longest to process data changes, as well as times of peak performance or strain on the system.

See also

- *Security Log Statistics* on page 398
- *Replication Statistics* on page 399

- *Messaging Statistics* on page 403
- *Messaging Queue Statistics* on page 407
- *Device Notification Statistics* on page 410
- *Package Statistics* on page 412
- *User Statistics* on page 414
- *Cache Statistics* on page 416

Data Change Notification History Statistics

Historical information for data change notifications (DCNs) consists of past notification details for monitored packages. Detailed data provides specific information on past notification activity for packages, and identifies which server data was affected.

Details about past notification events are organized into these categories:

Category	Description
Domain	The domain to which the package affected by the DCN belongs.
Package	The name of the package containing data changes.
MBO	The name of the MBO to which the notification applied.
Notification Time	The date and time that SAP Mobile Server received the DCN.
Processing Time	The time that SAP Mobile Server used to process the DCN.

Data Change Notification Performance Statistics

Data change notification (DCN) performance statistics consist of key performance indicators that reflect the efficiency of notification processing by SAP Mobile Server.

Performance monitoring highlights key totals and identifies average, minimum, and maximum values for primary activities. These calculations are dynamic, and are based on the data currently available in monitoring database for the specified time period.

All values in this table (totals, averages, maximums, minimums) apply to the specific time period you indicate:

Key performance indicator	Description
Total Notifications	The total number of notifications sent by the enterprise information system to SAP Mobile Server.
Average/Minimum/Maximum Processing Time	The average, minimum, or maximum amount of time SAP Mobile Server took to process a DCN.
Time at Minimum/Maximum Message Processing Time	The time of day at which the shortest or longest DCN processing event completed.

Key performance indicator	Description
Time of Last Notification Received	The time at which the most recent DCN was received by SAP Mobile Server.
MBO with Minimum/Maximum Notification Processing Time	The name of the package and associated mobile business object (MBO) with the shortest or longest notification processing time.

Device Notification Statistics

Device notification statistics provide data about the occurrence and frequency of notifications sent from SAP Mobile Server to replication synchronization devices.

Historical device notification monitoring reports on the packages, synchronization groups, and devices affected by replication payload synchronization requests in a given time frame. Performance-related device notification data provides a general indication of the efficiency of notification processing and the total demand of synchronization requests on the system.

See also

- *Security Log Statistics* on page 398
- *Replication Statistics* on page 399
- *Messaging Statistics* on page 403
- *Messaging Queue Statistics* on page 407
- *Data Change Notification Statistics* on page 408
- *Package Statistics* on page 412
- *User Statistics* on page 414
- *Cache Statistics* on page 416

Device Notification History Statistics

Historical information for device notifications provides specific information on past device notifications, indicating which packages, synchronization groups, and devices were involved in synchronization requests.

Details about past device notification events fall into these categories:

Category	Description
Application ID	The ID associated with the application.
Domain	The domain to which the package affected by the device notification belongs.
Package	The name of the package containing data changes.

Category	Description
Synchronization group	The synchronization group that the package belongs to.
Application Connection ID	The ID number of the connection participating in the synchronization request.
Generation time	The date and time that SAP Mobile Server generated the device notification.
User	The name of the user associated with the device ID.

Device Notification Performance Statistics

Device notification performance statistics provide a general indication of the efficiency of notification processing and the total demand of synchronization requests on the system.

Performance monitoring highlights key totals and identifies average, minimum, and maximum values for primary activities. These calculations are dynamic, and are based on the data currently available in monitoring database for the specified time period.

All values in this table (totals, averages, maximums, minimums) apply to the specific time period you indicate:

KPI	Description
Synchronization Group for Maximum Notifications	The synchronization group for which the maximum number of notifications were sent.
Package for Maximum Notifications	The package for which the greatest number of device notifications were sent.
Total Notifications	The total number of device notifications sent from SAP Mobile Server to devices.
Total Distinct Users	The total number of users that received device notifications. This value comprises only individual users, and does not count multiple synchronizations requested by the same user.
Total Distinct Devices	The total number of devices that received device notifications. This is distinct from Total Distinct Users, because a single user name can be associated with multiple devices.
Enabled Subscriptions	The total number of replication subscriptions for which notifications are generated.
Time at Last Notification	The time at which the last device notification was sent by SAP Mobile Server.

KPI	Description
Outstanding Subscriptions	The total number of replication subscriptions, both enabled and disabled.

Package Statistics

Package statistics reflect response times for replication-based and messaging-based synchronization packages.

This type of monitoring uses key performance indicators to provide data on the efficiency of response by SAP Mobile Server to synchronization requests. These calculations are dynamic, and are based on the data currently available in monitoring database for the specified time period.

See also

- *Security Log Statistics* on page 398
- *Replication Statistics* on page 399
- *Messaging Statistics* on page 403
- *Messaging Queue Statistics* on page 407
- *Data Change Notification Statistics* on page 408
- *Device Notification Statistics* on page 410
- *User Statistics* on page 414
- *Cache Statistics* on page 416

Replication Package Statistics

Replication package statistics consist of key performance indicators (KPIs) that reflect the overall function of the application environment at the cluster or domain level. The statistics highlight key totals and identify average, minimum, and maximum values for primary activities.

These key indicators monitor replication packages:

Note: These KPIs are not applicable at the MBO level.

- Total Bytes Received
 - Total Bytes Sent
 - Total Operation Replays
 - Total Devices
 - Total Rows Sent
 - Total Rows Received
-

KPI	Description
Total Devices	The total number of devices involved in synchronization. This total includes the same user multiple times if he or she has multiple devices. The value comprises individual devices, and does not count multiple synchronizations requested by the same device.
Total Rows Sent	The total number of rows sent during package synchronization.
Total Rows Received	The total number of rows received during package synchronization.
Total Errors	The total number of errors that took place across all synchronizations.
Total Bytes Received	The total number of bytes uploaded from clients to SAP Mobile Server.
Total Bytes Sent	The total number of bytes downloaded by clients from SAP Mobile Server.
Average/Minimum/Maximum Synchronization Time	The average, minimum, or maximum amount of time SAP Mobile Server took to finish a complete synchronization.
Time at Minimum/Maximum Synchronization Time	The time at which the shortest or longest synchronization completed.
Total Synchronization Requests	The total number of sync requests initiated by a client.
Total Operation Replays	The total number of operation replays performed by clients on MBOs.

Messaging Package Statistics

Messaging package statistics consist of key performance indicators (KPIs) that reflect the overall function of the application environment at the cluster or domain level. The statistics highlight key totals and identify average, minimum, and maximum values for primary activities.

Note: The Total Subscription Commands KPI is not applicable at the MBO level.

These key indicators monitor messaging packages:

KPI	Description
Total Subscription Commands	The total number of subscription commands sent from clients to the server.

KPI	Description
Total Devices	The total number of devices involved in synchronization. This total includes the same user multiple times if he or she has multiple devices. The value comprises individual devices, and does not count multiple synchronizations requested by the same device.
Average/Minimum/Maximum Processing Time	The average, minimum, or maximum amount of time SAP Mobile Server took to respond to a synchronization request message.
Time at Minimum/Maximum Messaging Processing Time	The time at which the shortest or longest response time completed.
Total Inbound Messages	The total number of messages sent from clients to SAP Mobile Server.
Total Outbound Messages	The total number of messages sent from SAP Mobile Server to clients.
Total Operation Replays	The total number of operation replays performed by clients on mobile business objects (MBOs).
Total Errors	The total number of errors that took place across all synchronizations.
Total Data Push	The total amount of data transmitted from the server to clients.

User Statistics

User statistics consist of key performance indicators that reflect the overall activity of application users.

User statistics can be filtered to include users who belong to a particular security configuration. This type of monitoring highlights key totals and identifies average, minimum, and maximum values for primary user activities. These calculations are dynamic, and are based on the data currently available in monitoring database for the specified time period.

Note: These statistics are not supported for SAP Mobile CRM and SAP Hybrid App for SAP application users.

See also

- *Security Log Statistics* on page 398
- *Replication Statistics* on page 399
- *Messaging Statistics* on page 403
- *Messaging Queue Statistics* on page 407
- *Data Change Notification Statistics* on page 408
- *Device Notification Statistics* on page 410
- *Package Statistics* on page 412

- *Cache Statistics* on page 416

Replication User Statistics

Replication user statistics reflect the synchronization activity of a group of replication-based synchronization users belonging to a specified security configuration. These statistics include general activity-related information on a per-user basis.

These key indicators monitor replication users:

KPI	Description
Total Synchronization Requests	The total number of sync requests initiated by a client.
Total Rows Received	The total number of rows received during package synchronization.
Total Rows Sent	The total number of rows sent during package synchronization.
Total Bytes Received	The total number of bytes uploaded from clients to SAP Mobile Server.
Total Bytes Sent	The total number of bytes downloaded by clients from SAP Mobile Server.
Average/Minimum/Maximum Synchronization Time	The average, minimum, or maximum amount of time SAP Mobile Server took to complete a synchronization request.
Time at Maximum/Minimum Synchronization Time	The time at which the fastest or slowest synchronization is completed.
Total Operation Replays	The total number of operation replays performed by user of mobile business objects (MBOs).
Total Errors	The total number of errors that took place across all synchronizations.
Total Devices	The total number of devices involved in synchronization. This total includes the same user multiple times if he or she has multiple devices. The value comprises individual devices, and does not count multiple synchronizations requested by the same device.

Messaging User Statistics

Messaging user statistics reflect the synchronization activity of a group of messaging-based synchronization users belonging to a specified security configuration. These statistics include general activity-related information on a per-user basis.

These key indicators monitor messaging users:

KPI	Description
Total Devices	The total number of devices involved in synchronization. This total includes the same user multiple times if he or she has multiple devices. The value comprises individual devices, and does not count multiple synchronizations requested by the same device.
Average/Minimum/Maximum Processing Time	The average, minimum, or maximum amount of time SAP Mobile Server took to respond to a sync request message.
Time at Minimum/Maximum Message Processing Time	The time of day at which the shortest or longest message processing event completed.
Total Inbound Messages	The total number of messages sent from clients to SAP Mobile Server.
Total Outbound Messages	The total number of messages sent from SAP Mobile Server to clients.
Total Operation Replays	The total number of operation replays performed by clients on mobile business objects (MBOs).
Total Errors	The total number of errors that took place across all synchronizations.
Total Subscription Commands	The total number of subscription commands sent from clients to the server.
Total Data Push	The total number of import data messages.

Cache Statistics

Cache statistics provide a granular view of cache activity either at the domain or package level, particularly in the areas of cache performance, mobile business object (MBO) status, and cache group status.

Cache statistics report on performance at the domain, package, MBO, and cache group levels to allow administrators to obtain different information according to the level of specificity required. These calculations are dynamic, and are based on the data currently available in monitoring database for the specified time period.

Note: These statistics are not supported for SAP Mobile CRM and SAP Hybrid App for SAP application users.

See also

- *Security Log Statistics* on page 398
- *Replication Statistics* on page 399
- *Messaging Statistics* on page 403
- *Messaging Queue Statistics* on page 407

- *Data Change Notification Statistics* on page 408
- *Device Notification Statistics* on page 410
- *Package Statistics* on page 412
- *User Statistics* on page 414

Cache Performance Statistics

Cache performance statistics report on key totals and identify average, minimum, and maximum values for primary cache activities. View cache performance data at the domain or package level.

Select either **Domain level** or **Package level** to view the following key performance indicators:

Key performance indicator	Description
Domain	The domain to which the package affected by the cache activity belongs.
Package	The name of the package associated with this cache activity.
Minimum/Maximum Cache Misses	The minimum or maximum number cache misses and the MBO name for which it was generated.
Minimum/Maximum Cache Hits	The minimum or maximum number of scheduled cache queries for all of the MBOs in the package in the specified date range.
Minimum/Maximum/Average % Cache Hits	The minimum or maximum percentage of scheduled cache queries for the supplied date range and the MBO name for which it was generated.
Minimum/Maximum Average Wait Time	The minimum or maximum average wait time for a scheduled cache query and the MBO name for which it was generated.
Minimum/Maximum Average Refresh Time	(Package-level only) The minimum or maximum average refresh time for an on-demand or scheduled refresh.

MBO Statistics

Mobile business object (MBO) status monitoring reports on cache activity at the MBO level, and thus, reflects activity for single mobile business objects.

Select **Package level MBO** to view the following key performance indicators:

Key performance indicator	Description
Cache Group	The name of the group of MBOs associated with this cache activity.

Key performance indicator	Description
MBO	The name of the single mobile business object associated with this cache activity.
Number of Rows	The number of rows affected by the cache refresh.
Cache Hits	The number of scheduled cache queries that occurred in the supplied date range.
Cache Misses	The number of on-demand cache or cache partition refreshes that occurred in the supplied date range.
Access Count	The number of cache queries that occurred in the supplied date range.
Minimum/Maximum/Average Wait Time	The minimum, maximum, or average duration of cache queries in the supplied date range. This time does not include the time required to refresh the cache in a cache “miss” scenario. Instead Minimum/Maximum/Average Full Refresh Time exposes this data.
Minimum/Maximum/Average Full Refresh Time	The minimum, maximum, or average duration of on-demand and scheduled full refresh activities in the supplied date range.

Cache Group Status Statistics

Cache group status statistics provide monitoring data about cache activity at the cache group level. The data reflects activity for all mobile business objects (MBOs) belonging to a cache group.

Select **Package level cache group** to view the following key performance indicators (KPIs):

KPI	Description
Package	The name of the package to which the associated cache group belongs
Cache Group	The name of the group of MBOs associated with the cache activity
Number of Rows	The number of rows in the cache table of the MBO
Last Full Refresh Time	The last time the cache or cache partition was fully refreshed
Last Update Time	The last time a row in the cache was updated for any reason (row-level refresh, full refresh, partitioned refresh, or data change notification)
Last Invalidate Time	The last time the cache was invalidated

KPI	Description
Cache Coherency Window	The data validity time period for the cache group, in seconds. Can span any value in this range: • 0 shows that data is always retrieved on-demand for each client. • 2049840000 shows that the cache never expires. This occurs when you set the on-demand cache group to NEVER expire or scheduled cache group to NEVER repeat.

Index

- Xmx maximum memory option 19
- XX:MaxPermSize permanent memory option 19

A

- administration
 - core administration nodes 2
- administration tasks
 - domain administrator 39
 - help desk operator 40
 - platform administrator 37
- administration users
 - configuring 104, 196
 - maintaining 104, 196
- administrators
 - domain administrator role 38
 - help desk role 39
 - platform administrator role 37
- Agentry applications 380
 - configuring Agentry Server 268
 - configuring logs 307
 - creating 267, 381
 - deploying 382
 - disabling 304
 - disconnecting active users 314
 - downloading metadata 383
 - enabling 304
 - rolling log files 161, 307
 - security configuration 247
 - starting/stopping/restarting server instance 293
 - viewing active users 305
 - viewing Agentry server instance information 305
 - viewing Agentry Server instance operation information 306
- Alert Message property 332
- alerts
 - effects of repository purging on history 32
- Alerts property 332
- alias, certificate 190
- Allow Roaming property 335
- Apache
 - Relay Server configuration 77
- APNS Device Token property 332

- Apple push notification properties 332
- application connection
 - activation options 322
- application connection properties 329
- application connection template 257, 260, 262, 266, 326–329
- application connection templates
 - administration overview 5
- application connections
 - administration overview 5
 - assigning Hybrid Apps to 321
 - reviewing package users for 324
- application ID
 - guidelines 320
- application log 155
 - registration log data 156
 - setting log data 156
- application settings 330
- application users 303
- applications 253, 254, 312
 - administration overview 5
- Application Connection properties 323
- audit destination 240
- audit filter 237
- audit formatter 241
- authentication cache timeout 200
- authentication failure 145
- authorization
 - DCNs 245
 - Push notifications 245
- auto purge
 - monitoring data 390
- automated message processing 334
- automatic registration 330

B

- backups
 - about 28
 - changing the schedule 29
 - forcing 30
 - restoring from 31
 - scheduling 28
 - suspending and resuming 29
- Badges property 332

Index

C

- cache 350
- cache database server pool size 106
- cache group
 - configuring 350
 - status statistics 418
- cache interval
 - real time 355
- cache monitoring 416
- cache performance statistics 417
- cache refresh
 - custom 353
 - daily 352
 - hourly 352
 - never schedule 355
 - on demand 351
 - scheduling 352
- cache statistics
 - viewing 397
- cache timeout, setting 200
- CDB 350
- certificate alias 190
- CertificateAuthenticationLoginModule
 - authentication module
 - for SAP single sign-on and X.509 221
- certificates
 - for context variables 375
 - managing for RSOE 88
- changing
 - cache database server pool size for SAP Mobile Platform 106
- cleaning up the Flash Player cache 14
- client application logs
 - checking 162
 - cleaning 163
- Client dispatcher 48, 49
 - configuring for HTTP client connections 50
- cluster configuration
 - configuration cache properties 58
 - performance properties 47
 - web container properties 56
- cluster properties 64
- clusters 41
 - administration overview 3
 - affected by configuration changes 41
 - cluster properties 41
 - status, checking 65
- communication ports
 - security configuration 51

- SSL encryption 51, 54
- configuration cache properties, configuring 58
- configuration changes
 - effect on clusters 41
- configuration files
 - Relay Server 72
- connection templates, creating 168
- Connections 116
- connections, creating 168
- console
 - about 33
 - commands 33
- context variables 376
 - configuring 375
- control flag 204
- controlFlag 204
- correlating log data across subsystems 136
- custom settings for messaging devices 334
- customization resource bundle
 - deleting from applications 302
- customization resource bundles
 - assigning to all existing application connections 300
 - assigning to application connections 300
 - deploying for applications 257, 263, 266, 299
 - description for applications 296
 - exporting from applications 301
 - guidelines and recommendations for applications 298
 - managing for applications 301
 - unassigning from applications 300
 - viewing for an application 301

D

- data cache
 - cache 350
- data change notification monitoring
 - histories 409
 - performance statistics 409
- data change notification statistics 408
- databases
 - monitoring 390
- DCN log data
 - general DCN 144
 - Hybrid App DCN 144, 145
- Debug Trace Level property 335
- Debug Trace Size property 335
- deleting 329
- Delivery Threshold property 332

- deploying Hybrid App packages 367
- deployment
 - Hybrid App archives 339
 - package archives 339
- Device Log Items property 335
- device notification
 - history statistics 410
 - performance statistics 411
- device notification log data 144
- device notification monitoring 410, 411
- device notifications
 - configuring 357
 - statistics 410
- Device Subtype property 335
- device users
 - assigning Hybrid App packages 377
- devices
 - Apple push notification properties 332
 - user assignments 315
- dispatcher log 154
 - dispatcher log data 154
 - service log data 155
- documentation roadmap 1
- DOE-C
 - invalid user 366
- DOE-C packages 364
- domain 105
 - security configuration, choosing 196
 - security, assigning security configuration 246
 - security, configuring 195
- domain administrator 38
- domain administrators
 - registering 103
- domain log data
 - DOE connections 147
 - JDBC connections 148
 - REST connections 149
 - SAP connections 150
 - SOAP connections 151
- domain log filter guidelines 123
- domain logs 110
- domain role mapping 197
- domains 100
 - administration overview 7
 - creating 101
 - deleting 102
 - enabling 101

E

- e-mail
 - redirecting with matching rules 373
- editing an application connection 323
- EIS
 - connection properties 170
 - Push operations 245
- Enable property 332
- enabling log profile 117
- enterprise information systems
 - See EIS
- environment variables
 - SCC_MEM_MAX 17, 19
 - SCC_MEM_PERM 17, 19
- error messages
 - logging levels 59
 - server logs 96
- errors
 - user account failure 366
- errors log data 146
- exporting log data 137

F

- Flash Player 15
 - cleaning up the cache after upgrade 14
- flush batch size for monitoring data 390
- flush threshold for monitoring data 390
- full backups 28

G

- getting started after installing 15
- GZIP 56

H

- hard coded credentials 376
- heat chart
 - launch icon 23
- help command (console) 33
- help desk 39
- help system, accessing 16
- HTTP 48, 49
 - configuring client dispatcher properties for 50
- HTTP compression 56
- HTTPS
 - RSOE certificates 88

Index

- Hybrid App 367
 - setting defaults 378
 - viewing default 325
- Hybrid App packages
 - assigning device users 377
 - configuring 372
 - configuring notification mailbox 371
 - deploying 368
 - deploying and managing 367
- Hybrid Apps
 - assigning to application connections 321
 - checking users and queues 379
 - configuring display name and icon 372
 - creating 258
 - Hybrid App packages administration 10

I

- icons
 - in SCC toolbar 23
- IIS
 - Relay Server configuration 77
- IMSI property 335
- incremental backups 28
- info command (console) 34
- iOS push notification properties 332

J

- Java system properties
 - displaying information about 34
- JDBC properties 171
- jvmopt memory options for Windows services 17, 19

K

- Keep Alive (sec) property 335
- keytool.exe 206

L

- layout for SAP Control Center views 27
- LDAP
 - configuration properties 207
 - stacking providers 205
- LDAP SSL configuration 206
- licenses
 - servers, reviewing 64

- log files
 - server logs 59
- Log files
 - rolling for Agentry applications 161, 307
- log filters 122
- log profile 117
- log, server
 - refreshing 98
- logging in to SAP Control Center - first user 15
- logging levels 59
- logging out of a server 17
- logical roles
 - DCNs 245
 - Push notifications 245
- login accounts, default
 - about 15
- login session timeout
 - setting 20
- logs
 - application 135
 - application logs 135
 - client application 162
 - dispatcher 134
 - dispatcher logs 134
 - DOE connections 130
 - domain-level 110
 - general DCN 127
 - Hybrid App DCN 127
 - JDBC connections 130
 - life cycles 59
 - outbound enabler 92
 - REST connections 130
 - SAP connections 130
 - SAP Mobile Server 96
 - server 96
 - server, configuring 59
 - SOAP connections 130
 - synchronization 125
 - synchronization logs 125

M

- managing Hybrid App packages 367
- Managing properties 257, 260, 262, 266, 328
- manual control of message processing 334
- Manual registration 254
- mapping roles
 - domain-level 197
- mapping roles for a package 342

- matching rules for redirecting e-mail
 - configuring 373
 - testing 374
 - MBO create error history 363
 - MBO data
 - See cache
 - MBO delete error history 363
 - MBO error history 363
 - MBO operation error history 363
 - MBO packages
 - contents, exporting 346
 - MBO status statistics 417
 - MBO update error history 363
 - memory
 - configuring 17
 - displaying information about 34
 - messaging 45
 - configuring properties 46
 - configuring subscriptions 359
 - messaging device advanced properties 335
 - messaging device connection properties 333
 - messaging devices
 - custom settings 334
 - information properties 335
 - messaging history monitoring
 - detail view 404
 - summary view 404
 - messaging monitoring
 - history 404
 - performance statistics 406
 - request statistics 403
 - messaging packages
 - statistics 413
 - messaging queues
 - statistics 407
 - status data 408
 - messaging statistics 403
 - messaging synchronization
 - monitoring 404
 - messaging users
 - monitoring 415
 - mobile business objects
 - cache group status statistics 418
 - clearing error history 363
 - reviewing error history 363
 - mobile devices
 - properties identifying 335
 - Model property 335
 - monitoring 394
 - cache 416
 - cache group status 418
 - cache performance 417
 - data change notification statistics 408
 - database, configuring 390
 - device notification history 410
 - device notification performance 411
 - device notifications 410
 - MBO status 417
 - messaging queue statistics 407
 - messaging statistics 403
 - messaging synchronization 404
 - messaging user statistics 415
 - replication statistics 399
 - replication user statistics 415
 - replication-based synchronization 400
 - statistic categories 397
 - user security 398
 - user statistics 414
 - monitoring data 394
 - auto purge 390
 - exporting 395
 - flush batch size 390
 - flush threshold 390
 - purging 395
 - reviewing 394
 - searching 396
 - monitoring profiles 392
 - creating and enabling 392
 - monitoring SAP Mobile Platform 387
 - overview 11, 388
 - monitoring schedule
 - custom 393
 - monitoring setup
 - effect on clusters 41
- ## N
- named security configuration
 - domain, selecting 196
 - native notifications
 - configuring 255, 259, 261, 265
 - Navigating applications 302
 - notification mailbox 371
 - notifications
 - configuring native 255, 259, 261, 265

Index

O

- Object API applications
 - creating 254
- OData applications
 - creating 264
- onboarding 5
- operation error history 363
- outbound enabler
 - configuring 78, 79, 89, 90
 - generate configuration file 77, 92
 - logging options 91
 - startup options 79, 90, 91
- Outbound Enabler 88
 - downloading log files 92
 - information in SAP Control Center 94
 - loading certificates 88
 - managing 92
 - setting up 78, 89

P

- package role mapping 342
- package statistics 412
- package subscriptions
 - configuring 359
 - managing 360
 - pinging 360
 - recovering 360
 - resuming 360
 - resynchronizing 360
 - suspending 360
 - unsubscribing 360
- package users
 - reviewing for an application connection 324
- packages
 - administration overview 8
 - cache properties 351
 - contents, importing 347
 - enabling and disabling 349, 365
 - Hybrid App administration overview 10
 - replication based synchronization 348
 - security 356
- payloads 117
- performance
 - SAP Control Center 14
- performance properties
 - configuring for cluster 47
- performance properties, configuring for server 50
- Perspective Resources view
 - about 24

- show/hide icon 23
- perspectives
 - about 24
 - creating 25
 - removing 25
 - renaming 25
- Phone Number property 335
- pinging a server 87
- platform administrators 37
- ports
 - displaying information about 34
- postinstallation tasks 15
- production edition 64
- profile definitions 112
- properties
 - advanced, of messaging devices 335
 - connection reference 170
 - custom settings for messaging devices 334
 - Hybrid Apps 372
 - information on messaging devices 335
 - monitoring database 390
 - package subscriptions, configuring 359
 - push notification for iOS 332
 - security provider configuration 207
- proxy 132
- Proxy 131
- proxy log
 - proxy log data 152
- proxy properties 192
- purging domain logs 138
- Push 131
- push log data 152
- push notification properties for iOS 332

Q

- queues
 - Hybrid App, checking 379
 - messaging, status data 408

R

- Refreshing application view 311
- registered application 105
- Relay Server
 - configuration properties in Relay Server tab 84
 - custom configuration 80
 - deleting configuration 83

- generate RSOE configuration 77, 92
 - information in SAP Control Center 83
 - managing 80
 - outbound enabler 88
 - properties, viewing or editing 80
 - Relay Server installation
 - configuring outbound enabler connection
 - settings 79, 90
 - configuring outbound enabler general
 - properties 78, 89
 - configuring outbound enabler start options 79, 90
 - configuring SAP Mobile Server to use Relay Server 72
 - custom configuration 74
 - defining server farms and cluster nodes 76, 82
 - generating outbound enabler configuration file 77, 92
 - generating Relay Server configuration file 77
 - launching Relay Server configuration wizard 74
 - modifying Relay Server configuration file 77
 - outbound enabler start options reference 91
 - quick configuration 72
 - reviewing configured Relay Server properties 77, 83
 - setting Relay Server general properties 74, 80
 - Relay Server Outbound Enabler 78, 88, 89
 - configuring 78, 79, 89, 90
 - downloading log files 92
 - generate configuration file 77, 92
 - information in SAP Control Center 94
 - loading certificates 88
 - logging options 91
 - managing 92
 - setting up 78, 89
 - startup options 79, 90, 91
 - Relay Server URL Suffix property 335
 - Relay Servers 71
 - replication history monitoring
 - detail view 400
 - summary view 400
 - replication monitoring
 - history 400
 - performance statistics 402
 - request statistics 399
 - replication packages
 - configuring subscriptions 357
 - statistics 412
 - replication statistics 399
 - replication subscription templates 357
 - replication synchronization 43
 - replication users
 - monitoring 415
 - replication-based synchronization
 - monitoring 400
 - repository 28
 - backing up 30
 - changing backup schedule 29
 - configuring purging 32
 - restoring from backup 31
 - scheduling backups 28
 - Request Response 132
 - resource explorer
 - launch icon 23
 - resources
 - remote, registering manually 21
 - REST API applications
 - creating 260
 - retrieving logs 120
 - role mapping
 - domain-level 197
 - package 342
 - RSOE 78, 89
 - rules for redirecting e-mail
 - configuring 373
 - testing 374
- ## S
- SAP
 - user account error 366
 - SAP audit measurement file 69, 70
 - SAP connection properties 187
 - SAP Control Center
 - about 1
 - connecting a browser to 15
 - console commands 33
 - functionality not applicable to SAP Mobile Platform 24
 - setting up SSL certificates 16
 - starting in Windows 13
 - starting in Windows as a service 13
 - stopping in Windows 13
 - SAP Control Center performance 14
 - SAP License Audit 69, 70
 - SAP Mobile Platform
 - configuring 41
 - monitoring 11, 388

Index

- SAP Mobile Platform administrators 37
- SAP Mobile Platform console
 - opening 20
- SAP Mobile Server
 - checking status 87
 - importing package contents 347
 - logging 96
 - logging out of 17
 - moving Hybrid App package contents from or to 346
 - moving MBO package contents from or to 346
 - pinging 87
 - server list 85
 - services provided by 84
 - stopping and starting 86
 - suspending and resuming 87
- SAP Mobile Servers
 - administration overview 5
- SAP SAP® Data Orchestration Engine Connector
 - connections 187
- SAP SAP® Data Orchestration Engine Connector
 - properties 187
- SAP single sign-on
 - deploying packages and bundles 364
 - SAPSSOTokenLoginModule authentication
 - properties 235
 - stacking login modules 204
- SAP single sign-on with X.509 CertificateAuthenticationLoginModule
 - authentication module 221
- SAP/R3 properties 183
- SAPSSOTokenLoginModule authentication
 - module
 - properties 235
- SCC_MEM_MAX 17, 19
- SCC_MEM_PERM 17, 19
- searching 312, 315
- Secure Sockets Layer encryption
 - communication ports 51
- secure synchronization port 43
- security
 - administration overview 11
 - domain, assigning security configuration 246
 - domain, configuring 195
 - monitoring 398
- security certificates
 - See SSL certificates
- security configuration
 - choosing 196
 - effect on clusters 41
 - packages 356
 - removing 196
- security configuration, creating 200
- security configurations
 - overview 199
- security log data 145
- security profile
 - communication port 51
 - management port 51
 - SSL certificates 52, 54
- security profiles 54
 - communication port 54
 - management port 54
- security provider configuration properties 207
- security providers, reordering 207
- server
 - status 87
- server compression 56
- server configuration 44
 - effect on clusters 41
 - system performance properties 50
- server licensing 64
- server log
 - deleting 98
 - searching 97
- server log data 153
- servers
 - log, refreshing 98
 - logging out of 17
 - logs, configuring 59
 - pinging 87
 - stopping and starting 86
 - suspending and resuming 87
- services
 - listing 34
- services, Windows
 - configuring SAP Control Center memory
 - options for 19
- shutdown command (console) 35
- SiteMinder authentication 247
- SiteMinder authentication cache timeout 251
- SiteMinder security configuration 249
- SiteMinder single sign-on 249
- SiteMinder SSO 249
- SLD: uploading payloads with SAP Control Center 67

- SOAP Web Services properties 190
 - Sounds property 332
 - SSL
 - mutual authentication 190
 - RSOE certificates 88
 - SSL certificates 16, 52, 54
 - setting up 16
 - SSL ciphers 54
 - SSL encryption
 - communication ports 51
 - security profile 54
 - SSL keystore 52
 - SSL truststore 52
 - stacking LDAP modules 205
 - stacking login modules
 - for SAP single sign-on 204
 - starting servers 86
 - statistics
 - application connection security 145
 - for messaging packages 413
 - for replication packages 412
 - statistics chart
 - effects of repository purging on 32
 - status command (console) 35
 - stopping servers 86
 - subscription templates
 - configuring for replication packages 357
 - creating 358
 - subscriptions, DOE-C
 - reviewing 366
 - SUP DCN User 245
 - SUP Push User 245
 - SupCertificateIssuer 377
 - SupCertificateNotAfter 377
 - SupCertificateNotBefore 377
 - SupCertificateSubject 377
 - SupPassword 376
 - for context variables 375
 - SupUser 376, 377
 - for context variables 375
 - sync group
 - configuring 349
 - synchronization
 - configuring general properties 43
 - synchronization listener properties 43
 - synchronization log
 - cache refresh 142
 - data services interface 143
 - data sync 139
 - operation replay 140
 - result checker 141
 - subscription 141
 - synchronization log data
 - cache refresh 139
 - data services interface 139
 - data synchronization 139
 - operation replay 139
 - result checker 139
 - subscription 139
 - synchronization port 43
 - system data, reviewing 394
 - System Landscape Directory (SLD)
 - registering destinations for 66
 - system licensing 64
 - system properties
 - displaying information about 34
- ## T
- timeout
 - setting for login sessions 20
 - toolbar icons 23
 - troubleshooting
 - authentication failure 145
 - Outbound Enabler 92
 - Relay Server Outbound Enabler 92
- ## U
- user interface, about 22
 - users 315
 - administration overview 5
 - administration, configuring 104, 196
 - administration, maintaining 104, 196
 - deleting 314
 - devices used by 315
 - Hybrid App, checking 379
 - messaging statistics 415
 - monitoring 414
 - security statistics 398
- ## V
- variables, context
 - configuring 375
 - view layouts, SAP Control Center
 - cascade 27
 - close all 27

Index

- horizontal tiling 27
 - minimize all 27
 - restore all 27
 - vertical tiling 27
 - Viewing applications 302
 - views
 - icons for managing 23
 - views, SAP Control Center
 - about 26
 - bringing to front of perspective 26
 - closing 26
 - maximizing 26
 - minimizing 26
 - opening 26
 - restoring 26
- ## W
- web container properties, configuring 56
- ## X
- XML audit file 69, 70