SYBASE®

An SAP® Company

Sybase Control Center for Online Data Proxy

Sybase Unwired Platform 2.1 ESD #3

DOCUMENT ID: DC01704-01-0213-01
LAST REVISED: May 2012

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

Contents

Get Started

Set up Sybase® Control Center.

About Sybase Control Center for Unwired Platform

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

Sybase Control Center combines a modular architecture, a rich administrative console, agents, common services, and tools for managing and controlling Sybase products. Unwired Platform is one of many Sybase products that use Sybase Control Center as it's management and administrative tool.

As part of an Unwired Platform installation, Sybase 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 Sybase 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 Sybase Control Center to set up an Unwired Server, manage packages, manage devices, configure mobile workflow packages, as well as review server and domain logs, and monitoring-related data.
- In a production environment, administrators use Sybase Control Center on a regular basis to perform the same tasks described for a shared development environment. They also configure the operation of Unwired 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 Unwired Platform

Sybase® Unwired 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. *Fundamentals* is available on the Sybase Product Documentation Web site.

Check the Sybase Product Documentation Web site regularly for updates: access *http://sybooks.sybase.com/nav/summary.do?prod=1289*, then navigate to the most current version.

Unwired Platform Administration by Node

The left navigation pane in the Sybase Control Center for Unwired Platform 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 Unwired Platform.

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

- There are two 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.
- You must have the correct Unwired Platform version and license for these nodes to be functional when they are visible. For example, Sybase Mobile Sales and Sybase Mobile Workflow products may not have all the same functionality as Sybase Unwired Platform.

Node	Purpose
Cluster	View general cluster properties and access the server list for the cluster, and Relay Server configurations on the cluster level. Configure a destination System Landscape Directory (SLD) server to deliver runtime information to a common SAP SLD repository. Export audit file that measures application usage to 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	View the list of servers, their properties, and their statuses. Expand this node to manage individual Unwired Servers, to configure properties and logs, and to apply pending changes.
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, 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.

Node	Purpose
Workflows	Deploy and manage mobile workflow packages and configure the notification mailbox. Deployed mobile workflow packages are listed below this node. Use the individual mobile workflow nodes to manage mobile workflow package properties, matching rules, context variables, error logs, application connections, and, optionally, queue items.
Monitoring	Create and manage settings for monitoring security, replication synchronization, messaging synchronization, device notification, data change notification, queue, package, user, and cache activities.

Server Administration Overview

The goal of server administration is to ensure that Unwired 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.

Table 1. Server administration tasks

Task	Frequency	Accomplished by
Installing the server	One-time installation per server	Unwired Platform installer.
Configuring the server to: • Set the replication and messaging synchronization ports, as well as communication ports for administration and DCN • Create security profiles for secure communication • Set up secure synchronization • Configure replication and messaging push notifications • Tune server performance	Postinstallation configuration with infrequent tuning as required	Sybase Control Center for Unwired Platform.
Manage the outbound enabler configuration for Relay Server. • Configure Relay Server properties • Manage certificates • View logs • Configure proxy servers for outbound enabler	Postinstallation	Sybase Control Center for Unwired Platform.
Setting server log file settings and subsystem log levels	Once, unless log data requirements change	Sybase Control Center for Unwired Platform.

Application and User Management Overview

The goal of application management is to register an application to Unwired 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, registration details, see *Administer > Applications* in Sybase Control Center online help.

Table 2. Application and user management tasks

Task	Frequency	Accomplish by using
Create new applications to register application entities with Unwired Server. A default application template is created automatically. Modify and delete applications as part of application life cycle.	As required	Sybase Control Center for Unwired Platform with Applications node, and Applications tab.
Create or modify application connection templates to specify details for native, workflow, and proxy application connections.	As required	Sybase Control Center for Unwired Platform, with Application node, and Application Connection Templates 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	Sybase Control Center for Unwired Platform, with Application 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	Sybase Control Center for Unwired Platform with the Application node, and Application Users tab.
Create a new activation code for a user whose code has expired.	As required	Sybase Control Center for Unwired Platform, with Application 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	Sybase Control Center for Unwired Platform with the Applications node.

Task	Frequency	Accomplish by using
Manage subscriptions	As required	Sybase Control Center for Unwired Platform with the Packages node.

Information and guidelines:

- Two activation options are available for onboarding, which refers to the process of activating an authentic device client, user (identified via name or email address), and application entity, as a combination, in Unwired Server:
  - Automatic activation – requires a user to present credentials to use the application on a supported device client.
  - Manual activation – requires the user to present the activation code upon log on to a supported device client. The system administrator establishes an activation code when registering the application connection for the user.
- Application templates are used for automatic activation. Therefore, when setting up the application template for automatic registration, be sure to set up the security configuration, domain, the application ID, and automatic registration enabled properties in application settings. Those are used for automatic application registration.

  When a client application connects to the server with its application ID and credentials, and requests automatic registration, the 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),the security configuration in the template is used to validate the credentials. Upon successful validation of those credentials, the user identity is registered in the Unwired Server. The client application may also include the security configuration as part of the username and in that case, the security configuration (in addition to application ID) is used to look up a matching template. If no or multiple templates are detected, the registration request fails. For details on how user names and security configuration names are processed when an email address is used, see *Considerations for Email Addresses as Username* in the *Security* guide.

Security Administration Overview

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

Unwired 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 Unwired Server to other components in the Unwired Platform must be protected.

Unwired 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 Unwired 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.

The platform administrator uses Sybase Control Center to monitor various aspects of Unwired 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.

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 3. System monitoring tasks

Task	Frequency	Accomplished by
Create and enable monitoring profiles	One-time initial configuration with infrequent tuning as required	Sybase Control Center for Unwired Platform with the Monitoring node
Enable domain logging	One-time setup with infrequent configuration changes, usually as issues arise	Sybase Control Center for Unwired Platform with the **Domains > <DomainName> > Log** node.
Review current/historical/ performance metrics	Routine	Sybase Control Center for Unwired Platform with the Monitoring node
Identify performance issues	Active	Sybase Control Center for Unwired Platform with the Monitoring node
Monitor application and user activity to check for irregularities	Active	Sybase Control Center for Unwired Platform with the Monitoring node
Troubleshoot irregularities	Infrequent	Reviewing various platform logs
Purge or export data	On demand	Sybase Control Center for Unwired Platform with the Monitoring node

Starting and Stopping Sybase Control Center in Unwired Platform

Sybase Unified Agent is used to start and stop Sybase Control Center.

There are two ways to start and stop the Sybase Control Center in an Unwired Platform environment.

- By default, SybaseControlCenter*X.X* is installed to run as a Windows service, and is set by the installer to start automatically.
- You can also use a command-line script as required.

- Start or stop from the Windows Control Panel; change automatic start and restart:
  a) Open the Windows Control Panel.

b) Select **Administrative Tools** > **Services**.

c) Locate SybaseControlCenter*X.X*. If the service is running, the status column displays "Started."

d) To start or stop the service, right-click the service and choose **Start** or **Stop**.

e) Double-click the service.

f) To set the service to automatically start when the system starts, change the **Startup type** to Automatic.

g) 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.

- Manually starting Sybase Control Center by command-line script:

  a) Enter the start command:

  ```
  <UnwiredPlatform_InstallDir>\SCC-X_X\bin\scc.bat
  ```

- Manually stopping Sybase Control Center by command-line script:

  a) Enter the stop command:

  ```
  <UnwiredPlatform_InstallDir>\SCC-X_X\bin\scc.bat -stop
  ```

  Note: You can use **scc.bat -stop** only to stop an SCC that was manually started with "scc.bat"; it cannot stop the SCC windows service.

Cleaning Up the Flash Player Cache

Sybase recommends you clean up the Flash Player cache, after upgrading to the latest version of Sybase Control Center. This is needed if you have used a previous version of Sybase Unwired 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`.
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.

Getting Started with Production Clusters

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

Note: For information about how to upgrade your installation to access cluster support for Online Data Proxy, see *Quick Start: Online Data Proxy*.

Getting Started After Installing

Perform postinstallation testing and configuration.

Prerequisites

Start Sybase Control Center.

Task

1. Install Adobe Flash Player 10.1 or later in the Web browser you will use to connect to Sybase 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 Sybase 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 Sybase 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 SCC login, and SUP login.
5. Learn about Sybase Control Center. To open the help system, click **?** in the upper-right corner of the screen, or select **Help** > **Online Documentation.**

Setting Up Browser Certificates for Sybase Control Center Connections

To avoid security exceptions when launching Sybase 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 Sybase Control Center installation.
- The browser session starts on the same computer as Sybase 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 b 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 Sybase 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:

   ```
   <UnwiredPlatform_InstallDir>\JDKX.X.X_XX\bin\keytool.exe -
   exportcert -alias jetty
   -keystore <UnwiredPlatform_InstallDir>\SCC-X_X\services
   \EmbeddedWebContainer\container\Jetty-X.X.XX\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 Sybase Control Center with an Installer-Defined Password

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

During installation, the person installing Unwired 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 Sybase Control Center.
2. Enter `supAdmin` for the user name and type the `<`*`supAdminPwd`*`>` for the password.
3. Click **Login.**

Logging out of Sybase Control Center

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

In order to protect system security, Sybase recommends that you log out of Sybase Control Center when you are not using the console.
Choose one of these methods:

- Click the **Logout** link at the top right corner of the console.

- From the Sybase Control Center menu, select **Application** > **Logout**.

Configuring Memory Usage

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

It is not usually necessary to configure memory usage for Sybase 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 • `jvmopt=-Xmx` – if you are running SCC as a Windows service • *SCC_MEM_MAX* – if you are starting SCC from the command line	• You need to prevent Sybase Control Center from using more than a given amount of memory • SCC fails to start and may display an error: `Could not create the Java Virtual machine.` • An OutOfMemory error says SCC is out of heap space • A warning message about system memory appears during the start process • The machine where SCC is installed has less than 2GB of memory. (Starting SCC on a machine with less than 2GB of memory triggers the startup warning message about system memory.)	On machines with less than 2GB of memory, set maximum memory to 256MB or more. Default value: none. (On machines with 2GB or more of memory, maximum memory is set dynamically and is effectively limited only by the amount of system memory available.)
Permanent memory • `jvmopt=-XX:MaxPermSize` – if you are running SCC as a Windows service • *SCC_MEM_PERM* – if you are starting SCC from the command line	An OutOfMemory error says SCC is out of permanent generation space	Increase by 32MB increments. If you reach a value equal to twice the default and still see the OutOfMemory error, contact Sybase technical support. Default value: 128MB

You can change memory options in two ways:

- For Sybase Control Center started from the command line – execute commands to set one or more environment variables before executing the **scc** command to start Sybase 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 Sybase Control Center service – modify a file used by the SCC service. When you use this method, your changes to the memory options persist—Sybase Control Center uses them every time it starts as a service.

Changing a Memory Option on the Command Line

Before you start Sybase 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 Sybase 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 Sybase Control Center.

Changing a Memory Option for an SCC Windows Service

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

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

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

   `<SCC-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 Sybase Control Center Windows service.

Configuring the Automatic Logout Timer

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

Prerequisites

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

Task

1. From the 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).

Manually Opening the Unwired Platform Console

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

Prerequisites

Before managing a cluster, ensure that the login has SCC administration privileges.

Task

1. In the SCC 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 Unwired 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 Unwired Platform cluster. Use this option if you have already mapped the SCC 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 SCC and Unwired Platform use different authentication repositories. Using different credentials in this step is unnecessary if SCC and Unwired 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 Unwired Platform console appears. If authentication fails, see *Sybase Control Center Issues* in the *Troubleshooting* guide.

Adding or Updating Unwired Server Registration Properties

By default a Sybase Control Center detects and registers clusters and Unwired Server nodes as managed resources of Sybase Control Center automatically: the resource entry named 'localhost' is created for the local server upon installation. However, you may need to manually register other new clusters or nodes or modify existing entries under specific conditions.

For information on these conditions, see *When Manual Managed Resource Property Changes Are Needed*.

1. Choose your action:
   - To register a new resource, on the Sybase Control Center menu, select **Resource > Register**.
   - To update the resource properties, on the Sybase 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 Unwired Server management 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 Sybase Control Center: click ***EntryName* > Manage** to launch the Unwired Platform management console.

When Manual Managed Resource Property Changes Are Needed

Understand the conditions under which managed resource properties need to be manually edited or added

These are the conditions under which you must manually create a new registration entry:

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

These are the conditions under which you must manually update an existing registration entry:

- If you modify the Unwired Server configuration to change the management port, you need to update these resource properties to match those values.

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

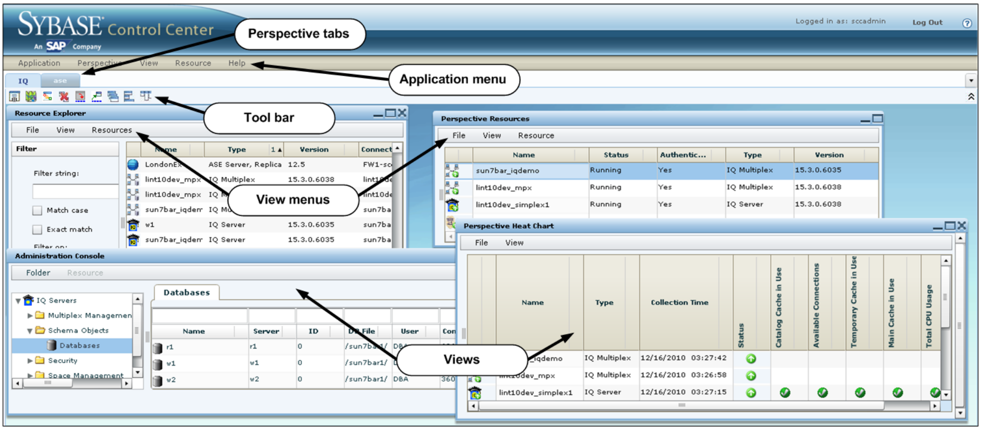
Understanding the Sybase Control Center Interface

Manipulate Sybase 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 Sybase Control Center user interface so you can identify them when they appear in other help topics.

Figure 1: Sybase Control Center User Interface



Name	Status	Authentic...	Type	Version
sun7bar_iqdemo	Running	Yes	IQ Multiplex	15.3.0.6035
lint10dev_mpx	Running	Yes	IQ Multiplex	15.3.0.6038
lint10dev_simplex1	Running	Yes	IQ Server	15.3.0.6038

Toolbar Icons

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

Table 4. 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).

Icon	Name	Description
	Launch Heat Chart	Opens the perspective heat chart, which gives a status overview of the registered resources in this perspective.
	Close All Open Views	Closes all open and minimized views.
	Minimize All 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.

Sybase Control Center Functionality Not Applicable to Unwired Platform

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

The following Sybase Control Center features can be disregarded in the context of Sybase Unwired Platform:

- Alerts
- Schedules
- Heat charts
- Historical performance monitoring
- Logging

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

Accessibility Features

This document is available in an HTML version that is specialized for accessibility. You can navigate the HTML with an adaptive technology such as a screen reader, or view it with a screen enlarger.

The Sybase CEP Option R4 documentation complies with U.S. government Section 508 Accessibility requirements. Documents that comply with Section 508 generally also meet non-U.S. accessibility guidelines, such as the World Wide Web Consortium (W3C) guidelines for Web sites.

For information about accessibility support in the Sybase IQ plug-in for Sybase Central™, see "Using accessibility features" in Chapter 1, "Introducing Sybase IQ" in *Introduction to Sybase IQ*. The online help for Sybase IQ, which you can navigate using a screen reader, also describes accessibility features, including Sybase Central keyboard shortcuts.

Note: You might need to configure your accessibility tool for optimal use. Some screen readers pronounce text based on its case; for example, they pronounce ALL UPPERCASE TEXT as initials, and MixedCase Text as words. You might find it helpful to configure your tool to announce syntax conventions. Consult the documentation for your tool.

For information about how Sybase supports accessibility, see Sybase Accessibility at *http://www.sybase.com/accessibility*. The Sybase Accessibility site includes links to information on Section 508 and W3C standards.

Sybase Control Center Accessibility Information

Sybase Control Center uses the Adobe Flex application.

For the most current information about Adobe Flex keyboard shortcuts, see *http://livedocs.adobe.com/flex/3/html/help.html?content=accessible_5.html*.

Note: To use Sybase Control Center with JAWS for Windows screen reading software effectively, download and install the appropriate Adobe scripts. See *www.adobe.com*.

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 Sybase 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.

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**.

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, Sybase Control Center displays one.

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**.

Views

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

In Sybase 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 persective.

Each perspective includes these views:

- Perspective Resources
- Administration Console

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, click a resource, pull down its menu using the handle 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.
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 **View** > **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.

Arranging View Layout in a Perspective

Use the view layout options to manage your perspective space.

Click one of these icons in the Sybase 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

Icon	Action
▤	**Tile All Open Views Horizontally**

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

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

Repository

The Sybase 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, Sybase Control Center saves backups as follows:

- Each full backup is stored in its own subdirectory in `<SCC-install-directory>/backup`.
- Each incremental backup is stored in a file in `<SCC-install-directory>/backup/incremental`.

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

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.

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.

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**.

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

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 Sybase 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 Sybase Control Center.
2. Copy the most recent full backup from `<SCC-install-directory>/backup/<generated_directory_name>` to `<SCC-install-directory>/services/Repository`. For example:

   ```
   copy C:\sybase\SCC-3_2\backup\repository.
   270110161105\scc_repository.db
   C:\sybase\SCC-3_2\services\Repository
   ```
3. If you have no incremental backups to load,
   a) Also copy the log file from `<SCC-install-directory>/backup/<generated_directory_name>` to `<SCC-install-directory>/services/Repository`. For example:

      ```
      copy C:\sybase\SCC-3_2\backup\repository.
      270110161105\scc_repository.log
      C:\sybase\SCC-3_2\services\Repository
      ```

   b) Skip to step *5* on page 23.

4. 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-install-directory>\services\Repository

   ..\..\bin\sa\bin_<platform>\dbsrv11.exe scc_repository -ad
   <SCC-install-directory>\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 Sybase Control Center.
   If you loaded incremental backups, SCC 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:\sybase\SCC-3_2\services\Repository

% ..\..\bin\sa\bin_windows32\dbsrv11.exe scc_repository -ad
C:\sybase\SCC-3_2\backup\incremental
```

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 Sybase 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 Sybase 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**.

Sybase Control Center Console

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

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

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

Console Commands

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

help Command

Display syntax information for one or more Sybase 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.

info Command

Display information about specified parts of the Sybase 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 Sybase 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 Sybase 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 Sybase 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 Sybase 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 Sybase Control Center server:

  ```
  info -p
  ```

Permissions

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

shutdown command

Stop the Sybase Control Center server if it is running.

Syntax

```
shutdown
```

Examples

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

  ```
  shutdown
  ```

Permissions

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

status Command

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

Syntax

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

Parameters

- **-a | --agent –** display the status of the Sybase Control Center agent component.
- **-h | --help –** display information about the **info** command.
- **-p | --plugin [*plugin-name*] –** display the status of the specified Sybase 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 Sybase 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 Repository
  ```

Permissions

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

Sybase Control Center Security

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

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

Administer

Use Sybase Control Center for Unwired 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, Unwired Platform administrators need to create a scalable IT infrastructure using clusters. Clustering creates redundant Unwired 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 Unwired Platform deployment typically uses at least one relay server. The connections to relay servers can be configured within a cluster instance from Sybase Control Center.

Cluster-Affecting Configuration Changes

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

When you make a cluster-affecting change on the primary Unwired 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:

- server configuration
- monitoring setup
- security configuration

Copying and Pasting Properties

Values displayed in property tables in Sybase 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).

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 Sybase Control Center navigation pane, click the name of the cluster.
2. In the administration view, click **General**.
3. 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.

Viewing Cluster Information

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

1. In Sybase 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 the name of the host computer upon which the primary Unwired Server node was installed with a `_cluster` suffix appended to the host name.
    - The number of servers and outbound enablers that are members of the cluster.
    - The cluster sync data shared path, if enabled.
    - The asynchronous operation replay queue count. See *Configuring Asynchronous Operation Replay Queue Count*.

Checking System Licensing Information

Review licensing information to monitor available and used 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 Unwired Platform. For more information on license types, see *Sybase Unwired Platform Licenses* in *System Administration*.
    - 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, Unwired Server generates a license expired error and Unwired 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.
- Total device license count – the total number of device licenses available with your license. This count limits how many devices can connect to your servers. See *Sybase Unwired Platform Licenses* topics in *System Administration* for licensing information.
- Used device license count – the total number of unique devices associated with the users currently registered with the server. If all of your available device licenses are in use, you can either upgrade your license or manually delete unused devices to make room for new users:
  - For workflow and Online Data Proxy client devices, delete the Application Connections that are no longer in use.
  - For native replication-based applications, delete Package Users on the respective Package.
  - For native messaging-based applications, delete the Application Connections associated with the clients not in use.

  See *System Administration* for licensing information.
- Device license expiry date – the date and time at which the device license expires. When a device license expires, Unwired Server generates a license expired error and connection requests from registered devices are unsuccessful.
- 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.
- Used application user license count – the number of all registered application users of all applications. This value represents the cumulative total of the distinct user identities registered for each application. The same user identity using:
  - Multiple versions of the same application counts as one application user.
  - Two different applications count as two application users.

4. Click **Close**.

Note: Unwired Platform licensing is configured during installation. However, if necessary, license details can be changed at a later time. See *Manually Updating and Upgrading License Files* in *System Administration*.

Checking Cluster Status

Verify that a cluster is running.

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

Sharing Cluster Information With SAP Servers

If your Unwired 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. Sybase Unwired Platform is a third-party system that must be registered with SLD.

To prepare the SLD server environment for Unwired Platform, ensure the following pre-requisites:

- 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 Sybase Unwired Platform must be the latest Common Information Model version (currently 1.6.21).

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 Sybase 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 Repeat Password	The user account password used to authenticate the user name entered. Password and Repeat 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 Unwired Platform. Deleting an SLD server removes it from Sybase Control Center and you can no longer use it as a payload destination.

1. In the navigation pane of Sybase 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 Sybase Control Center

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

Prerequisites

Task

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 Sybase 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 Unwired Platform Architecture* in the *System Administration* guide.

1. In the navigation pane of Sybase 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 **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.
    - 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.

   b) Click **OK**.

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

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 Sybase Control Center, click the cluster name.
2. From the menu bar of the **System Landscape Directory** page, click **Schedule**.
3. Click **Disable**.

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 Sybase Unwired Platform cluster.

Use Sybase Control Center for Unwired 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: Unwired Server Management API*.

Sharing Application Data with SAP License Audit

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

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

Generating the SAP Audit Measurement File

Use Sybase Control Center to generate an audit measurement file.

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

1. In Sybase Control Center, select the Unwired 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 Sybase 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* .

Relay Server

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

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

Each Unwired 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 Unwired Server. Connections between the RSOE and Relay Server use HTTPS protocol.

Relay Server also provides load balancing for the Unwired Server cluster by forwarding requests from client devices to Unwired 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 Unwired Server to use Relay Server, using these high-level steps::

1. Use Sybase Control Center to configure an Unwired 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 Sybase 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 *Installation*

Guide for Runtime or visit *http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.help.sqlanywhere.12.0.1/relayserver/relayserver12.html* for Relay Server installation and configuration information.

3. Set up Outbound Enablers on each Unwired Server node.

Configuring Unwired Server to use Relay Server

Choose a method for configuring Unwired Server to use Relay Server, then generate a Relay Server configuration file. Copy the file to the Relay Server host, and quickly distribute the same configuration to multiple Relay Server nodes, with minimal changes.

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

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 *Installation Guide for Runtime*.

Creating a Quick Configuration

Create a Relay Server configuration primarily with system defaults, and create Outbound Enabler (RSOE) processes for each Unwired Server.

1. In the navigation pane, click the Unwired Server cluster name.
2. In the administration pane, click the **Relay Servers** tab.
3. Click **Quick Configure**.
4. Specify these property values:
    - **Host –** for single Relay Server environments the host name of the Relay Server. Or, in the case of a Relay Server farm environments, the host name of the load balancer.
    - **Http port –** the Relay Server HTTP port.
    - **Https port –** the Relay Server HTTPS port.
    - **URL suffix –** the URL suffix used by the Outbound Enabler to connect to a Relay Server. The prefix you set depends on whether Relay Server is installed on IIS or Apache hosts. For IIS, you would use `/ias_relay_server/client/rs_client.dll`. For Apache you would use `/cli/iasrelayserver`.
    - **Replication or Messaging 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); this same token can be shared by replication or messaging farms. The replication and messaging farm token values can be the same.
    - **(Optional) Description –** a user-definable description of the Relay Server.
5. (Optional) Check **Advanced settings** and specify these property values:

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

6. (Optional)For Online Data Proxy deployments, 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 RSOE authentication on the Internet proxy server.
   - **Http proxy password –** password for RSOE authentication on the Internet proxy server.
7. Click **OK** to generate a Relay Server configuration file, and the RSOE processes for each Unwired Server.

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

Six RSOE instances (three each for messaging and replication ports) are created on each Unwired Server host, but they not started.

Next

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

Creating a Custom Relay Server Configuration

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

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 Unwired 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. Specify these property values:

- **Host –** for single Relay Server environments the host name of the Relay Server. Or, in the case of a Relay Server farm environments, the host name of the load balancer.
- **Http port –** the Relay Server HTTP port.
- **Https port –**

  Note: If Relay Server uses HTTPS and certificates, clients other than replication may not be able to connect: messaging applications only support HTTP, and hybrid workflow container applications for iOS support HTTPS — but not certificates.

  the Relay Server HTTPS port.
- **URL suffix –** the URL suffix used by the Outbound Enabler to connect to a Relay Server. The prefix you set depends on whether Relay Server is installed on IIS or Apache hosts. For IIS, you would use `/ias_relay_server/client/rs_client.dll`. For Apache you would use `/cli/iasrelayserver`.
- **(Optional) Description –** a user-definable description of the Relay Server.

2. Add or remove HTTP credentials as required:
   a) Check **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**.

Reviewing Configured Relay Server Properties

Review the Relay Server configuration to confirm property values, before you generate the configuration file.

1. Confirm the property values, ensuring that:
   - No errors exist.
   - All Unwired Server clusters are defined, and the correct type.
2. Click **Finish**.

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

Next

When you have finished adding all required Relay Servers, set up one or more Outbound Enabler (RSOE) for each Unwired Server in the cluster.

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 after export.

1. In the navigation pane, click the Unwired Server cluster name.
2. In the administration pane, click the **Relay Servers** tab.
3. Click **Generate**.
4. Choose **Relay server properties configuration file**.
5. Select the parts of the file to generate:
   - The entire Relay Server configuration
   - A server node definition
   - A farm definition
6. Select an output target for the file.
7. Click **Finish**.
8. Manually edit the file if this is required, and save the changes.
   For example, Relay Server farms that use a load balancer will likely require further edits. Consider:
   - Updating the host name if the Relay Server needs to share the same name of the load balancer.
   - Adding Relay Servers to the farm for which the load balancer manages inbound requests.

   For details on other manual edits that you can perform, see the Relay Server documentation at *http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.help.sqlanywhere.12.0.1/relayserver/relayserver12.html*.
9. If you need 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.

Next

If you are deploying multiple Relay Servers in a farm with a third-party load balancer, all Relay Server Outbound Enablers must be configured to connect to the load balancer host name as well. When all Relay Servers and Outbound Enablers are configured, test the setup and ensure that connectivity from Unwired Server is possible.

Setting Up RSOE

Set up one or more RSOEs for each Unwired 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> > Server Configuration**.
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 or Messaging protocol.
   - **Unwired sever port –** select the port on which RSOE will manage requests.
   - **Relay server host –** for single Relay Server environments the host name of the Relay Server. Or, in the case of a Relay Server farm environments, the host name of the load balancer.
   - **Relay server port –** select the Relay Server HTTPS port.
   - **Unwired server farm –** select the string that identifies the Unwired 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 Unwired Server in the cluster. This property is case-sensitive, and it 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. Specify these property values:
   - **Http user name –** 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.
2. 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.
3. Specify these property values:

- **Certificate file –** select this option and choose the `.CRT` file used to authenticate the RSOE to relay server. You can only choose this file if you have already loaded it into the Unwired Server certificate store.
- **Trusted certificate –** if the certificate file includes multiple certificates, choose whether to trust a single certificate or all of them.

Configuring RSOE Start Options

Configure start options for RSOE.

1. Enable an option:
   a) Check the box that corresponds to each name.
   b) Set a value. If you check the box but set no value for the option, the default is used.
2. Click **OK**.
3. Ensure the process starts by checking the Status column of the Outbound Enablers tab.

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 Sybase 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 Unwired Server hosts.

1. In the navigation pane, click the Unwired Server cluster name.
2. In the administration pane, click the **Relay Servers** tab.
3. Click **Generate**.
4. Choose **Outbound enabler configuration XML file**, then click **Next**.
5. Select an output target for the file.
6. Click **Finish**.

Managing Configured Relay Servers

Relay Servers configured with Sybase Control Center are registered in the Unwired Server cluster database. Administrators can view or edit configuration properties, and delete Relay Servers in Sybase 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.

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 Unwired 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. Specify these property values:
   - **Host –** for single Relay Server environments the host name of the Relay Server. Or, in the case of a Relay Server farm environments, the host name of the load balancer.
   - **Http port –** the Relay Server HTTP port.
   - **Https port –**

     Note: If Relay Server uses HTTPS and certificates, clients other than replication may not be able to connect: messaging applications only support HTTP, and hybrid workflow container applications for iOS support HTTPS — but not certificates.

     the Relay Server HTTPS port.
   - **URL suffix –** the URL suffix used by the Outbound Enabler to connect to a Relay Server. The prefix you set depends on whether Relay Server is installed on IIS or Apache hosts. For IIS, you would use `/ias_relay_server/client/rs_client.dll`. For Apache you would use `/cli/iasrelayserver`.
   - **(Optional) Description –** a user-definable description of the Relay Server.
2. Add or remove HTTP credentials as required:
   a) Check **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**.

Define Server Farms and Cluster Nodes

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

Repeat these steps to add or remove multiple Unwired Server clusters, as needed.

1. Define the Unwired Server cluster.
   a) Specify these property values:
      - **Farm ID –** a string that identifies the Unwired Server cluster, for which the Relay Server manages requests. This property is case-sensitive, and it must match the value in the Outbound Enabler configuration.
      - **Type –** the type of request managed by the Relay Server, Replication or Messaging protocol.
      - **(Optional) Description –** user-definable description of the Unwired Server cluster.

   b) Click **+**.
   c) Repeat steps 1 and 2 to add multiple Unwired Server clusters.
   d) To delete a configured Unwired Server cluster, select it in the list, then click the **X** button.
2. Identify each Unwired Server instance in the cluster.
   a) Select an existing Unwired Server cluster.
   b) Specify these property values:
      - **Node ID –** a string that identifies the Unwired 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); this same token can be shared by replication or messaging farms.

   c) Click **+**.
   d) Repeat steps 1 and 2 to add Unwired Server cluster nodes.
   e) To delete a configured Unwired Server node, select it in the list and click **X**.
3. Click **Next** to review your settings, or click **Finish** to exit the wizard.

   Note: After each change, you may need to update the relay server configuration, and take steps to manually update the relay server configuration.

Reviewing Configured Relay Server Properties

Review the Relay Server configuration to confirm property values, before you generate the configuration file.

1. Confirm the property values, ensuring that:

- No errors exist.
- All Unwired Server clusters are defined, and the correct type.

2. Click **Finish**.

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

Next

When you have finished adding all required Relay Servers, set up one or more Outbound Enabler (RSOE) for each Unwired Server in the cluster.

Deleting a Relay Server Configuration

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

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

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 Unwired Server cluster name.
2. In the administration pane, click the **Relay Server** tab.
3. Select a Relay Server.
4. Click **Refresh**.

Relay Server Tab Reference

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

Column	Description
Host	for single Relay Server environments the host name of the Relay Server. Or, in the case of a Relay Server farm environments, the host name of the load balancer.
Http port	the Relay Server HTTP port.
Https port	the Relay Server HTTPS port.

Column	Description
URL suffix	the URL suffix used by the Outbound Enabler to connect to a Relay Server. The prefix you set depends on whether Relay Server is installed on IIS or Apache hosts. For IIS, you would use `/ias_relay_server/client/rs_client.dll`. For Apache you would use `/cli/iasrelayserver`.

Unwired Server

The Unwired Platform runtime server is called Unwired Server. Unwired Server manages the data exchange process between the enterprise and device clients to create a homogeneous layer in a diverse mobile ecosystem. In a production environment, the Unwired Server must be installed on a 64-bit host.

Unwired 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 – 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.

Server List

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

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 Sybase Control Center are registered automatically. Once registered, remote servers also appear in the server list.

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

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 Sybase Control Center for servers that are installed on the same host as Sybase 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 Sybase Control Center for Unwired 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.

1. In the Sybase Control Center navigation pane, click **Servers** to display the servers list.
2. Select one or more servers 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**.
   - 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.

Starting and Stopping RSOE

Start and stop an RSOE process as needed. All configured RSOEs are started by default when the Unwired Server starts.

1. In the navigation pane, click **Servers > *ServerNode* > Server Configuration**.
2. In the administration pane, select the Outbound Enabler tab, select the RSOE instances, and click **Start** or **Stop**.

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 Unwired Server in order to enable the suspend and resume server functions.

Task

1. In the Sybase Control Center navigation pane, click **Servers** to display the servers list.
2. Select one or more servers in this list.
3. Choose an appropriate process:
   - To suspend the server, click **Suspend**. Wait for about 1 minute, and click "refresh" button. When the server status changes from `suspend pending` to `suspended`, you can then perform the administration actions you required.
   - To then 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 Unwired 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, stopped, suspended, suspend pending, or resume pending.
4. Use the controls in the administration console to start, stop, restart, suspend, or resume the server, as required.

Server Properties

Server properties let administrators manage server configuration settings to ensure smooth data exchange between the server and client. You can configure administration port, replication, and messaging properties in the Server Configuration node of Sybase Control Center.

Note: Properties you configure for an Unwired Server are cluster-affecting. Therefore, to make sure they are propagated correctly, Sybase recommends that you set them only on a primary cluster server.

General Server

Configure properties and security profiles for Unwired Server management and communication ports. These ports process incoming replication synchronization, administration, and data change notification requests. You must secure data transmission over management and DCN communication ports by creating and assigning an SSL configuration to the ports. You can also configure Unwired Server performance properties.

Unwired Server Management Ports

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

If you choose an unencrypted administration channel, simply reconfigure the port and change the profile used. You must then ensure that the managed resource properties Sybase Control Center uses for outbound requests match that of Unwired Servers.

Note: If you change the profile from one that requires server authentication only (for example, the built-in profile named **default**) to one that requires mutual authentication (for example, as the built-in profile named default_mutual), you must have already saved the necessary SSL certificates to both the Unwired Server and the Sybase Control Center keystores and truststores. See *Security* for details.

Additionally, if you are using Sybase Control Center in a development/test environment, you must also configure the Unwired WorkSpace on all development computers accessing the development Unwired Server to also save the required certificates to the java keystore. Sybase does not recommend that you use mutual authentication for the management port.

Configuring Security Profiles

Configure security profiles to secure communication between Unwired Server administration and DCNs.

Prerequisites

Before creating a security profile, ensure that you possess digital certificates that have been verified and signed by third-party trusted authorities, as well as import required certificates in to the Unwired Server keystore.

Task

1. In the left navigation pane, expand the **Servers** folder and select a server.
2. Select **Server Configuration**.
3. In the right administration pane, select the **General** tab.
4. From the menu bar, select **SSL Configuration**.
5. Create a new Security Profile:

   a) Name the security profile.
   b) Enter the case sensitive certificate alias for the profile (defined in the server keystore).
   c) Select the Authentication option.
6. Click **Save**.
7. In the server restart dialog, click **OK**.
8. Restart the server for these changes to take effect.

Next

Use the profile to encrypt administration and DCN ports.

Configuring SSL Properties

Configure SSL certificates and security profiles to facilitate Secure Sockets Layer (SSL) encryption for communication ports in Unwired Platform.

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.

Task

Note: If you enable SSL for one node in a cluster, you must enable it for all other nodes, since the secure connection is established at the cluster level. The port numbers for each node are not dependent; that is, they can be identical or unique.

Defining Certificates for SSL Encryption

For the primary server, specify keystore and truststore certificates to be used for SSL encryption of Unwired Platform communication ports. All security profiles use the same keystore and truststore.

For secondary Unwired Servers, SSL properties are synchronized from the primary server. Therefore for secondary servers, these properties are still visible, but cannot be edited.

1. In the left navigation pane, expand the **Servers** folder and select a server.
2. Select **Server Configuration**.
3. In the right administration pane, select the **General** tab.
4. From the menu bar, select **SSL Configuration**.
5. To configure SSL encryption for all security profiles, complete these fields:
   - **Keystore Location –** the full path name indicating the location where the keys and certificates are stored. Certificates used for administration and data change notification ports are stored in the keystore. The path should be relative to `<Unwired`

  `Platform_InstallDir>\UnwiredPlatform-XX\Servers\UnwiredServer`.
- **Keystore Password –** the password that secures the key store.
- **Truststore Location –** the full path name for the public key certificate storage file. The Certificate Authority (CA) certificates used to sign certificates store their public keys in the truststore. The path should be relative to `<Unwired Platform_InstallDir>\UnwiredPlatform-XX\Servers\UnwiredServer`.
- **Truststore Password –** the password that secures the truststore.

Note: If at any point you have changed the password for the keystore and truststore with keytool, then you must remember to update the password here as well.

6. Click **Save**.

Next

Create an SSL security profile that uses the selected certificates.

Creating an SSL Security Profile in Sybase 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. Unwired Server uses multiple listeners. Clients that support the same characteristics can communicate to Unwired Server via the same port defined in the listener.

Note: A security profile can be used by one or more servers in a cluster.

1. In the left navigation pane, expand the **Servers** folder and select a server.
2. Select **Server Configuration**.
3. In the right administration pane, select the **General** tab.
4. From the menu bar, select **SSL Configuration**.
5. In the **Configure security profile table**:
   a) Enter a name for the security profile.
   b) Enter a certificate alias. This is the logical name for the certificate stored in the keystore.
   c) Select an authentication level:

      If the security profile authenticates only the server, then only the server must provide a certificate to be accepted or rejected by the client. If the security profile authenticates both the client and the server, then 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.

Profile	Authenticates	Cipher suites
intl	server	• SA_EXPORT_WITH_RC4_40_MD5 • RSA_EXPORT_WITH_DES40_CBC_SHA
intl_mutual	client/server	• RSA_EXPORT_WITH_RC4_40_MD5 • RSA_EXPORT_WITH_DES40_CBC_SHA
strong	server	• RSA_WITH_3DES_EDE_CBC_SHA • RSA_WITH_RC4_128_MD5 • RSA_WITH_RC4_128_SHA
strong_mutual	client/server For example, this is the required option for mutual authentication of Unwired Platform and Gateway.	• RSA_WITH_3DES_EDE_CBC_SHA • RSA_WITH_RC4_128_MD5 • RSA_WITH_RC4_128_SHA
domestic	server	• RSA_WITH_3DES_EDE_CBC_SHA • RSA_WITH_RC4_128_MD5 • RSA_WITH_RC4_128_SHA • RSA_WITH_DES_CBC_SHA • RSA_EXPORT_WITH_RC4_40_MD5 • RSA_EXPORT_WITH_DES40_CBC_SHA • TLS_RSA_WITH_NULL_MD5 • TLS_RSA_WITH_NULL_SHA

Profile	Authenticates	Cipher suites
domestic_mutual	client/server	• RSA_WITH_3DES_EDE_CBC_SHA • RSA_WITH_RC4_128_MD5 • RSA_WITH_RC4_128_SHA • RSA_WITH_DES_CBC_SHA • RSA_EXPORT_WITH_RC4_40_MD5 • RSA_EXPORT_WITH_DES40_CBC_SHA • RSA_WITH_NULL_MD5 • RSA_WITH_NULL_SHA

6. Click **Save**.
7. From the **Communication Ports** menu, assign the security profile to the desired management or communication ports.

Next

If you configure a secure port on one server, you must enable it on every node in the cluster, then restart all servers in the cluster to commit the configuration changes.

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 server 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 an Unwired Server when configuring SSL.

1. To enable OCSP when doing certificate revocation checking, check **Enable OCSP**.
2. Configure the responder properties (location and certificate information):

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

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 Unwired Server Performance Properties

To optimize Unwired Platform performance, configure the thread stack size, maximum and minimum heap sizes, user options, and inbound and outbound messaging queue counts.

1. In the left navigation pane, expand the **Servers** folder and select a server.
2. Select **Server Configuration**.
3. In the right administration pane, select the **General** tab.
4. From the menu bar, select **Performance Configuration**.
5. Configure these replication payload properties, as required:
   - Host Name – the name of the machine where Unwired Server is running (read only).
   - Thread Stack Size – the JVM `-Xss` option.
   - Minimum Heap Size – the minimum size of the JVM memory allocation pool, in megabytes. For production recommendations on this value, see *Unwired Server Replication Tuning Reference* in *System Administration*.

- Maximum Heap Size – the maximum size of the JVM memory allocation pool, in megabytes. For production recommendations on this value, see *Unwired Server Replication Tuning Reference* in *System Administration*.
- User Options (in Show optional properties) – other JVM options. For example, you can enable JVM garbage collection logging by setting `-XX:+PrintGCDetails`. Or you can set the permanent space which allocated outside of the Java heap with `DJC_JVM_MAXPERM`; the maximum perm size must be followed by K, M, or G, for example, `-XX:MaxPermSize=512M`. Note that DJC_JVM_MAXPERM is not visible to Sybase Control Center.

6. For messaging payloads, click the **Messaging** tab and configure these properties, as required:

   - Maximum Number of In Memory Messages – specify the number of in-memory messages to allow.
   - Inbound Messaging Queue Count – the number of message queues used for incoming messages from the messaging-based synchronization application to the server. Sybase recommends a choose a value that represents at least 10% of active devices.
   - Outbound Messaging Queue Count – the number of message queues used for outbound messages from the server to the messaging-based synchronization application. Sybase recommends a choose 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.
   - Subscribe Bulk Load Thread Pool Size – the maximum number of threads allocated to initial bulk load subscription operations. The default value is five. Setting the thread pool size too high can impact performance.

   Note: If you increase either queue count property, ensure you also increase the MaxThread property in the `<hostname>_iiop1.properties` file.

7. Click **Save**.

Saving and Refreshing an Unwired Server Configuration

Refreshing an Unwired Server configuration displays the latest effective configuration information.

After successfully saving a server configuration, refresh the configuration to display the most recent updates. To commit these changes to the server, restart the server before saving subsequent updates. The refresh function must be used in conjunction with a server restart for the displayed configuration to be applied.

If you refresh the configuration in between two sets of saved configuration changes without injecting a server restart following the refresh, only the second set of changes are committed and consequently displayed as the current set of properties used by Unwired Server.

Note: Follow the steps in exactly the order they appear. Otherwise, configuration changes will be lost.

1. Reconfigure Unwired Server as required.
2. Click **Save**.
3. Click **Refresh** to display original values; the recent'y saved changes are not displayed.
4. Restart Unwired Server to commit those changes, using the method you prefer for server restarts.
5. In the left navigation pane, expand the **Servers** folder and select a server.
6. Select **Server Configuration**.
7. In the right administration pane, select the appropriate tab and click **Refresh**.
   Current server configuration properties committed with the restart action appear.
8. Make the next set of configuration changes, as required.

Reviewing Pending Changes

As you configure Unwired Server with Sybase Control Center, changes that require a server restart are aggregated to the **Pending Changes** tab for the server name you are currently administering.

Changes listed in this window require a server restart before they take effect.

1. In the left pane, click the Unwired server you are currently logged into.
2. Click **Pending Changes**.
3. Review all listed changes that are pending.
4. If the changes are valid, click **Restart** to commit the changes.
5. A confirmation message to continue appears. Confirm that you want to restart the server.
6. Review Unwired Server status messages on the **General** tab to ensure that the server has restarted and changes have been committed successfully. If the update is successful, the bolded text and asterisk (*) are also removed from the respective server name in the left navigation pane.

Applying Multiple Unwired Server Configuration Changes

A server restart writes the changes made in Sybase Control Center to the appropriate Unwired Server configuration file. To apply multiple server configuration changes with a single server restart, you cannot make consecutive conflicting updates or refresh the configuration in between saved changes.

Consider these important points when applying multiple changes to an Unwired Server configuration:

- Failure to save a configuration change prior to restarting Unwired Server results in configuration changes being lost.

- Failure to restart Unwired Server after saving a configuration change results in changes being uncommitted; Unwired Server instead uses the values that currently exist in the configuration file (that is, previous configuration properties and values).
- Cumulative saved changes are applied successfully upon server restart as long as these updates do not conflict. Attempting to save two conflicting sets of changes fails. In this case, inject a server restart in between each saved change to ensure that the required updates are propagated across the server.
- Refreshing the server configuration displays the latest successfully saved configuration information. If you click Refresh in between two sets of saved changes, only the most recent saved updates are applied during a server restart.

When you must make multiple changes to the same component of the Unwired Server configuration, follow this procedure:

Note: Follow the steps in exactly the order they appear. Do not use the Refresh function in between saved changes. Otherwise, configuration changes will be lost.

1. Make the first set of configuration changes, as required.
2. Click **Save**.
   A confirmation message appears in the administration console indicating the success or failure of the save.
3. Make the second set of configuration changes, as required.
4. Click **Save**.
   A confirmation message appears in the administration console indicating the success or failure of the save. If the save is unsuccessful, restart the server before reattempting these updates.
5. Restart Unwired Server to commit the changes in steps 1 and 3, using the method you prefer for server restarts.
6. In the left navigation pane, expand the **Servers** folder and select a server.
7. Select **Server Configuration**.
8. In the right administration pane, select the appropriate tab and click **Refresh**.
   Current server configuration properties committed with the restart action appear.

Viewing Unwired Server Properties

View information, including host names, port numbers, version, and file location, to help you manage an Unwired Server and its components.

1. In the left navigation pane, expand the **Servers** folder and select a server.
2. In the right administration pane, select the **Properties** tab.
3. Review Unwired Server properties.

Relay Server Outbound Enabler

The Outbound Enabler (RSOE) runs as an Unwired Server process and manages communication between the Unwired 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 Unwired Server on its Replication or Messaging port. Unwired Server sends its response to the RSOE, which forwards it to the Relay Server, to be passed to the client.

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

Note: Sybase 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 Unwired 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*** **> Server Configuration**.
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 Unwired 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> > Server Configuration**.
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 or Messaging protocol.
   - **Unwired sever port –** select the port on which RSOE will manage requests.
   - **Relay server host –** for single Relay Server environments the host name of the Relay Server. Or, in the case of a Relay Server farm environments, the host name of the load balancer.
   - **Relay server port –** select the Relay Server HTTPS port.
   - **Unwired server farm –** select the string that identifies the Unwired 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 Unwired Server in the cluster. This property is case-sensitive, and it 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. Specify these property values:
   - **Http user name –** 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.
2. 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.
3. Specify these property values:

- **Certificate file –** select this option and choose the `.CRT` file used to authenticate the RSOE to relay server. You can only choose this file if you have already loaded it into the Unwired Server certificate store.
- **Trusted certificate –** if the certificate file includes multiple certificates, choose whether to trust a single certificate or all of them.

Configuring RSOE Start Options

Configure start options for RSOE.

1. Enable an option:
   a) Check the box that corresponds to each name.
   b) Set a value. If you check the box but set no value for the option, the default is used.
2. Click **OK**.
3. Ensure the process starts by checking the Status column of the Outbound Enablers tab.

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 Sybase 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 Unwired Server hosts.

1. In the navigation pane, click the Unwired Server cluster name.
2. In the administration pane, click the **Relay Servers** tab.
3. Click **Generate**.
4. Choose **Outbound enabler configuration XML file**, then click **Next**.
5. Select an output target for the file.
6. Click **Finish**.

Managing Configured RSOEs

Manage RSOE instances you have configured.

Retrieving RSOE Logs

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

1. In the navigation pane, click **Servers > <ServerNode> > Server Configuration**.
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.

Viewing or Editing Individual RSOE Properties

View or edit configuration properties for a selected RSOE instance.

Relaunching the RSOE Configuration Wizard

Relaunch the Outbound Enabler Configuration wizard to create a new RSOE configuration.

1. In the navigation pane, click **Servers > <ServerNode> > Server Configuration**.
2. In the administration pane, select the Outbound Enabler tab.
3. Select the RSOE instance, then click **Properties**.

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> > Server Configuration**.
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 or Messaging protocol.
   - **Unwired sever port –** select the port on which RSOE will manage requests.
   - **Relay server host –** for single Relay Server environments the host name of the Relay Server. Or, in the case of a Relay Server farm environments, the host name of the load balancer.
   - **Relay server port –** select the Relay Server HTTPS port.
   - **Unwired server farm –** select the string that identifies the Unwired 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 Unwired Server in the cluster. This property is case-sensitive, and it 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. Specify these property values:
   - **Http user name –** 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.

2. 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.
3. Specify these property values:
    - **Certificate file –** select this option and choose the `.CRT` file used to authenticate the RSOE to relay server. You can only choose this file if you have already loaded it into the Unwired Server certificate store.
    - **Trusted certificate –** if the certificate file includes multiple certificates, choose whether to trust a single certificate or all of them.

Configuring RSOE Start Options

Configure start options for RSOE.

1. Enable an option:
    a) Check the box that corresponds to each name.
    b) Set a value. If you check the box but set no value for the option, the default is used.
2. Click **OK**.
3. Ensure the process starts by checking the Status column of the Outbound Enablers tab.

Outbound Enabler Start Options Reference

Review available Outbound Enabler start options. These options affect Outbound Enabler logging. Each Outbound Enabler has its own log file that you can retrieve in Sybase 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 for technical support.
Reconnect delay	5	Delay before retry after connection fails.
Maximum output file size	10KB	Maximum log file size.
Truncate log file	None	Option to delete the log file at RSOE startup.
Advanced	None	User-defined value for start parameters. For more information on configuring RSOE options, see *Outbound Enabler* in *SQL Anywhere 12.0.1* at http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.help.sqlanywhere.12.0.1/relayserver/ml-relayservers-6039420.html

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> > Server Configuration**.
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*.

Deleting RSOE Configurations

Delete an RSOE configuration to remove the configuration properties from the cluster database.

1. In the navigation pane, click **Servers > <ServerNode>** > **Server Configuration**.
2. In the administration pane, select the Outbound Enabler tab, and select the RSOE instances.
3. Stop the RSOE instances and click **Delete**.
4. Click **OK**.

Refreshing the RSOE List

Refresh the RSOE list to display current information about deployed and configured RSOE instances.

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

Starting and Stopping RSOE

Start and stop an RSOE process as needed. All configured RSOEs are started by default when the Unwired Server starts.

1. In the navigation pane, click **Servers >** ***ServerNode*** **> Server Configuration**.
2. In the administration pane, select the Outbound Enabler tab, select the RSOE instances, and click **Start** or **Stop**.

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> > Server Configuration**.
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**.

Outbound Enabler Tab Reference

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

Column Name	Displays
Server Node ID	the string that identifies the Unwired Server in the cluster. This property is case-sensitive, and it must match the value in the Relay Server configuration.
Unwired Server Port	the port on which Outbound Enabler manages requests.
Farm Type	the type of request managed by the Relay Server, Replication or Messaging protocol.
Unwired Server Farm	the string that identifies the Unwired 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	for single Relay Server environments the host name of the Relay Server. Or, in the case of a Relay Server farm environments, the host name of the load balancer.
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 Unwired Server, retrying... –** Check the Unwired 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 Unwired 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: `Relay Server outbound enabler is running. Please check the log file to confirm the status.` 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). Sybase 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 Unwired Server certificate store.

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

Server Log

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

These server logs are available:

- Unwired Server logs – collect data on Unwired Server health and performance by component, and retrieve data for all or specific searches. You can save and archive system logs, and manage log file size and rollover behavior.
- Messaging Server logs – create trace configurations for messaging modules, and retrieve trace data for all or specific messages. Export data for archive or further analysis.

Note: Properties you configure for an Unwired Server are cluster-affecting. Therefore, to make sure they are propagated correctly, Sybase recommends that you set them only on a primary cluster server.

Unwired Server Runtime Logging

Unwired Server logs collect runtime information from various embedded runtime components.

By default, all the components of the Unwired Server log are set at the INFO level, except the Other components, which are set at the WARN level. However, you can change this level as required. You should only use Sybase Control Center to set these logging values to ensure they are configured correctly. These values will be correctly transcribed to an internal file (that is, `<UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\UnwiredServer\Repository\logging-configuration.xml`).

You can view these Unwired Server logs in two ways:

- From Sybase Control Center – click **Servers** > ***primaryServer*** > **Log** in the left pane, and **Unwired Server** > **General**in the right 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.
- From a text editor – browse to and open one or more of the `<UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\UnwiredServer\logs\<hostname>-server.log` files. These files may be indexed, depending on how you configure the life cycle for the servers's log file.

Configuring Unwired Server Log Settings

Configure Unwired 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. Server log properties are only set on the primary node; property settings on secondary nodes are read-only.

How changes are applied in a cluster depends on whether you are configuring a primary or secondary server. Sybase recommends you only configure log settings on the primary server. If you change the setting on a secondary server, the configuration is updated only for that server and is temporary (eventually the primary settings are propagated to all servers in the cluster).

Additionally, you should always use Sybase 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 Sybase Control Center left navigation pane, click **Servers > *primaryServer* > Log**, and in the right pane click **Unwired Server > Settings**.
2. 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.
3. Set the 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.
4. For each component, choose a log level:

Component	Default Log Level
MMS	Info
PROXY	Info
MSG	Info
Security	Info
Mobilink	Info
DataServices	Info
Other	Warn

Component	Default Log Level
DOEC	Info

Note: DOEC only appears if you run the DOEC installer after installing SUP cluster.

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 Unwired Server is running in non-service mode)
Off	Do not record any messages

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

Log life cycle default example

If you keep the default maximum file size and default index, an Unwired 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.

Viewing the Unwired 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**, **Last**, and **Go to** controls. Use **View Details** open the actual log file and find the corresponding line.

There are also two search options you can use:

- Basic search – allows you to search by keyword, log level, first/last X number of lines in the log file.
- Advanced search – allows you to 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 Unwired Server Log Data

Filter server log data according to the criteria you specify.

1. In the Sybase Control Center left navigation pane, click **Servers >** ***primaryServer*** **> Log**, and in the right pane click **Unwired Server > General**.
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, Sybase 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 Server Log Settings* in Sybase Control Center online help.
   - **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 Unwired Server Log

Update the information in the log console window.

1. In the Sybase Control Center left navigation pane, click **Servers >** ***primaryServer*** **> Log**, and in the right pane click **Unwired Server > General**.
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 Unwired Server Log

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

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

Messaging Server Runtime Logging

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.

You can configure trace settings for the primary server cluster in Sybase Control Center for each module. The settings are available to cluster servers through the shared data folder.

Configuring Messaging Server Log Settings

Configure trace configuration properties for modules to specify the amount of detail that is written to the log. Messaging Server log settings are cluster-wide, so changes made on the primary node are effective on all nodes.

How changes are applied in a cluster depends on whether you are configuring a primary or secondary server. Sybase recommends you only configure log settings on the primary server. If you change the setting on a secondary server, the configuration is updated only for that server and is temporary (eventually the primary settings are propagated to all servers in the cluster).

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 Sybase 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 Sybase Control Center left navigation pane, click **Servers > *primaryServer* > Log**, and in the right pane click **Messaging Server > Settings**.
2. 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 Unwired 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.
SUPBridge	This module handles communications from the messaging server to the Unwired Server (what are the official terminology?). 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.

3. 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: `<UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\UnwiredServer\logs`.

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 Sybase Control Center left navigation pane, click **Servers >** ***primaryServer*** **> Log**, and in the right pane click **Messaging Server > General**.
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 Sybase Control Center left navigation pane, click **Servers >** ***primaryServer*** **> Log**, and in the right pane click **Messaging Server > General**.
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 Sybase Control Center left navigation pane, click **Servers >** ***primaryServer*** **> Log**, and in the right pane click **Messaging Server > General**.
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 Unwired Server. The returned date and time is the Unwired 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 Unwired 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 Sybase Control Center from the Unwired Platform administration perspective of Sybase Control Center.
- You can support multiple customers inside the same Unwired 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 Unwired 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.

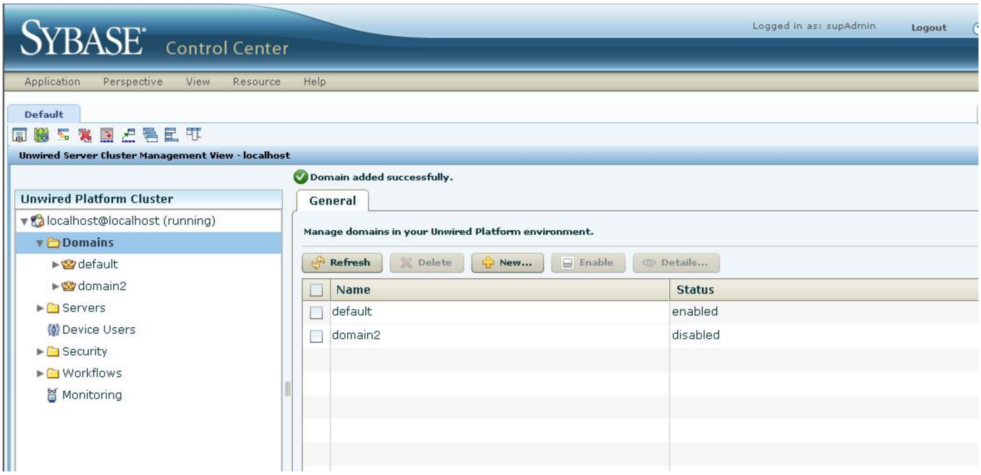
The "default" domain

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. Sybase recommends creating new domains to facilitate tenants according to their application requirements.



Domain added successfully.

Manage domains in your Unwired Platform environment.

	Name	Status
□	default	enabled
□	domain2	disabled

Creating and Enabling a New Domain

Create and configure multiple domains within a single Unwired 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**.
4. Optional. Select a security configuration for the domain by checking an option from the list of available configurations. These security configurations are then available for use in validating users accessing the packages.
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.

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**.

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 Sybase 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.

- (Optional) **Company name** – the name of the organization the administrator belongs to. Sybase recommends you supply this information if you are setting up Unwired 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.

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.

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.

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 Sybase 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.

Scheduling Domain-Level Cleanup

Periodically clean up accumulated data maintenance items in cache that are no longer needed.

You can automate domain-level cleanup based on a configured schedule for specific cleanup categories.

Running the cleanup options uses system resources, so Sybase recommends that you schedule these tasks when system load is lightest. Optionally you can run the cleanup tasks manually.

1. In the Sybase Control Center left navigation pane, expand the **Domains** tab and select a domain.
2. In the right pane, select the **Scheduled Task** 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 not active for the 'number of inactive days' in the schedule task configuration. Note, subscription is considered active as follows: • Replication – last synchronization request timestamp. • Messaging – last synchronization message timestamp.
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. **Note:** Only error messages are removed.
Client Log Cleanup	Removes client log records that have already been synchronized to the device, or are no longer associated with active users.
Synchronization Cache Cleanup	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. This cleanup task also removes unused or stale partitions.

4. Select **Enable**. Schedules run until you disable them, or they expire.

Scheduling Cleanup Options

The SUP Administrator or SUP Domain Administrator schedules domain-level data maintenance cleanup.

Set up an automatic schedule for database cleanup:

1. In the left pane, select the cluster, then the domain.
2. In the right pane, select the **Scheduled Tasks** tab.
3. Select one of the cleanup options:

Option	Description
Subscription Cleanup	Removes subscriptions that are not active for the 'number of inactive days' in the schedule task configuration. Note, subscription is considered active as follows: • Replication – last synchronization request timestamp. • Messaging – last synchronization message timestamp.
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. **Note:** Only error messages are removed.
Client Log Cleanup	Removes client log records that have already been synchronized to the device, or are no longer associated with active users.
Synchronization Cache Cleanup	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. This cleanup task also removes unused or stale partitions.

4. Click **Properties**.
5. In the Task Properties dialog, select the **Schedule** tab, and set the appropriate options:
   - **Schedule repeat** – select how often the schedule should run. Options are **hourly**, **daily**, **custom**, and **never.**
     - If you select **hourly** or **daily**, specify:
       - **Start date** – select the date and time the automated cleanup should begin. Use the calender picker, and 24-hour time selector.
       - **End date** – select the date and time the automated cleanup should end.
       - **Days of the week** – select each day the automated cleanup schedule should run.
     - If you select **custom**, you can specify the interval granularity by seconds, minutes, or hours, as well as other date and time parameters.
     - If you select **never**, no scheduling options are available.
6. In the Task Properties dialog, select the **Options** tab, set the number of inactive days for which to purge.

   Note: This step is unnecessary for Synchronization Cache Cleanup.

7. Click **OK** to save the schedule properties and purge options.

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 pane, select the cluster, then the domain.
2. In the right pane, select the **Scheduled Task** tab.
3. Select one of the cleanup options, and verify the value in the Status column is set to **disabled**.
4. On the **Scheduled Task** tab, click **Enable**.
5. Click **OK** 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 pane, select the cluster, then the domain.
2. In the right pane, select the **Scheduled Task** tab.
3. Select one of the cleanup options, and verify the value in the Status column is set to **enabled**.
4. On the **Schedule** tab, click **Disable**.
5. Click **OK** to confirm. The value in the Status column changes to **disabled**.

Running Manual Purge by Domain

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

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

1. In the left pane, select the cluster, then the 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 number of days for which to preserve data. Artifacts that fall outside of the time period are purged.

5. Click **OK** to confirm. The request is sent immediately, and the task runs asynchronously on Unwired 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 Log Profiles* in *Sybase Control Center* online help.

Enabling Application Logging

Enable domain-level logging to help you trace and monitor application activities, and review the resulting logs for troubleshooting.

Creating and Enabling Domain Logging

Create logging profile definitions and enable the log profile.

1. In the left navigation pane of Sybase Control Center, select **Domains**.
2. Under the domain node, select **Log**.
3. In the right administration pane, select the **Settings** tab.
4. Click **New**.
5. In the Profile Definition dialog, enter a **Name** and **Description** for the log profile.
6. Add the necessary profile definitions.
7. Select **Enable after creation**.
8. Click **OK**.

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

   1. Under the **Settings** tab, click **Configuration**.
   2. Check **Enable flush threshold**.

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.

Adding or Removing Applications to the Profile

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

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**.

Adding or Removing Security Configurations or Users 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 packages 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**.

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 applications currently deployed in this domain.
3. You can do any of the following:
   - To add a connection to a new or existing profile, select the check-box adjacent to the **Connection Type** entry in the list and select the connection names.

     Note: To add connections for proxy services, select the **Proxy** connection type.

- To remove an application from the profile, select the **View selection** check-box. Select the connection type and click **Remove**.

4. Click **OK**.

Adding or Removing Application Connections to the Profile

Add application connections to the profile. This enables you to list current application connections to Unwired 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**.

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 SUP 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.

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**.

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:

   1. In the **Settings** tab, select the log profile you have created.
   2. Click **Enable**.
   3. Click **OK** on the confirmation dialog.

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 Unwired 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 Sybase Control Center, select **Domains**.
2. Under the 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.
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 configuration** 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.

Subsystem	Description
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 workflow 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, RES, SAP®, and SOAP connections.
Proxy	Provides a view of Online Data Proxy connection-related activities. Within this subsystem, categories include request response, and push.

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 Sybase Control Center to create and manage your filters.

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.

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 Sybase 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. Select **+** to add a filter definition to a subsystem.
5. In the Filter Definition dialog, enter the **Name** and **Description** of the filter.
6. Select the **Sub System**.

7. 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.

8. Click **OK**.

Deleting Filters

Delete the filters created for sub systems

1. In the left navigation pane of the Sybase 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. Select the filter from the list.
5. Click **Delete**.

Updating Filters

Update filters as needed to fine tune log file filtering.

1. In the left navigation pane of the Sybase 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. Select the filter from the list.
5. 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.

6. In Filter Definition, modify the description, and set up the filter criteria.
7. Click **OK**.

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 **Domain > *<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, and setting a **Time Order** (ascending or descending).

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. Click **Retrieve** to retrieve the logs.
   The table is populated with the list of logs.

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, and setting a **Time Order** (ascending or descending).

   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.

Retrieving Errors 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, and setting a **Time Order** (ascending or descending).

   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.

Retrieving Proxy Request-Response 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** tab and then select **Request Response**.
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, and setting a **Time Order** (ascending or descending).

   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.

Retrieving Proxy Push Logs

Retrieves the log data for all push notifications 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** tab and then 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, and setting a **Time Order** (ascending or descending).

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.

Retrieving Client Logs

An administrator can retrieve the client logs as a text file by making modifications to a configuration file.

1. In the installation directory, navigate to the location `<UnwiredPlatform_InstallDir>\Server\UnwiredServer\config`.
2. In the `OnlineProxy.properties` file, set the value of DUMP_CLIENT_LOGS property to `True`.
3. Restart the server for the configuration changes to take effect.
   The client log is saved as a text file in the location: `<UnwiredServer_InstallDir>\Servers\UnwiredServer\logs\OnlineProxy\`.

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 ProxyRequestResponse tab and enter the filter information; retrieve the data; then select Correlated mode, and switch to another tab such as ProxyPush.

1. In the left navigation pane of Sybase 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 **Correlated mode** and 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.

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. Sybase 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 Sybase 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. Click **Retrieve** to retrieve the logs.
   The table is populated with log data.
8. Click **Export** and specify the file name and location.

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 Sybase 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 Unwired Server interface. Using data in these logs and the correlation tool, you can follow the data path between the enterprise information system (EIS), Unwired 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 datasource	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 Unwired 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 – the name of the user associated with the application ID.
- 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.

Note: Additional detail columns:

- Payload

Operation Replay

Synchronization logs include data related to different aspects of data synchronization, including operations and Unwired Server interface.

Operation Replay – statistics for moving MBO operations (typically create, update, and delete) from the device cache to the cache database cache on Unwired 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 – the name of the user associated with the application ID.
- 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.

Note: Additional detail columns:

- Payload

Subscription

Synchronization logs include data related to different aspects of data synchronization, including subscriptions and Unwired Server interface.

Subscription – statistics for transferring data between mobile devices and the cache database on Unwired 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 – the name of the user associated with the application ID.
- 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.

Note: Additional detail columns:

- Payload

Result Checker

Synchronization logs include data related to different aspects of data synchronization, including result checker and Unwired 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 – the name of the user associated with the application ID.
- Stage – the current stage of processing - START or FINISH.
- Package – the name of the package to which the subscription belongs.
- Class – 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.

Note: Additional detail columns:

- None

Cache Refresh

Synchronization logs include data related to different aspects of data synchronization, including cache refresh and Unwired 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 – the name of the user associated with the application ID.
- 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.

Note: Additional detail columns:

- Refresh Type
- Virtual Table Name
- Partition Key
- Pre Update Cache Image
- Post Update Cache Image

DS Interface

Synchronization logs include data related to different aspects of data synchronization, including data service and Unwired 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 – the name of the user associated with the application ID.
- 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.

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 Unwired 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 – the name of the user associated with the application ID.
- 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.

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 workflow 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 – the name of the user associated with the application ID.

- 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.

Note: Additional detail columns:

- Payload

Workflow Data Change Notification

Provides logging data for workflow data change notifications between an enterprise information system (EIS) and an MBO package.

- Time – the time and date stamp for the log entry.
- Workflow ID – the unique identifier associated with a workflow.
- Application Connection ID – the unique identifier for a user application connection.
- User – the name of the user associated with the application ID.
- 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 workflow DCN request subject line.
- From – the "From" value for the workflow DCN request.
- To – the "To" value for the workflow DCN request.
- Body – the message body for the workflow DCN request.
- Error – the error message if any.

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 – the name of the user associated with the application ID.
- 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.
- Method – 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.

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 – the name of the user associated with the application ID.
- 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.

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 Unwired Server, or Unwired 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 (payload).
- Domain – the core and administrative (MMS) domain name (payload).
- JSON Message Content – the messaging synchronization JSON message (payload). This is the SUP-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 (payload).
- 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 – the name of the user associated with the application ID.
- 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.

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 (payload).
- Domain – the core and administrative (MMS) domain name (payload).
- 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 – the name of the user associated with the application ID.
- 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.

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 (payload).
- Domain – the core and administrative (MMS) domain name (payload).
- 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 – the name of the user associated with the application ID.
- 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.

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 (payload).
- Domain – the core and administrative (MMS) domain name (payload).
- 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 – the name of the user associated with the application ID.
- 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.

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 (payload).
- Domain – the core and administrative (MMS) domain name (payload).
- 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".

Proxy Log

Proxy log data includes data for all requests and responses from the Proxy server, and all push notifications from the Proxy server.

Proxy Request-Response Log

Proxy Request-Response log data includes data for all requests and responses made 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 – the name of the user associated with the application ID.
- 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.
- 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.

Note: Additional detail columns:

- Post Data

- Request Header Fields
- Response Body
- Response Header Fields

Proxy Push Log

Proxy push logs include log data for all push notifications 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 – the name of the user associated with the application ID.
- 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.

Note: Additional detail columns:

- Message Body

Connections

Connections allow Unwired 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 establish, Unwired Server can open and close it as required.

Connection Pools

Unwired 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 Unwired 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 Unwired Server, connection pools are based on an existing template created for a specific data source type.

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.

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:
   - Proxy - choose this if you care connecting to the Online Data Proxy.
6. Select the appropriate template for the data source target from the **Use template** menu. By default, several templates are installed with Unwired Platform; however, a production version of Unwired 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 you add.

Note: In a remote server environment, if you edit the sampledb 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.
9. Click **OK** to register the connection pool.
   The name appears in the available connection pools table on the Connections tab. Administrators can now use the connection pool to deploy packages.

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 create a new 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, click the **Connections** icon.
2. Select the **Connection Pool Name** you want to validate.
3. Click **Properties**.
4. 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 create a new 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 Sybase Control Center to enterprise information systems (EIS) .

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
User	Specifies the user name for HTTP basic authentication, if applicable.	User name
Certificate Alias	Sets the alias for the Unwired Platform keystore entry that contains the X.509 certificate for Unwired 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 Unwired Server certificate keystore.

Name	Description	Supported Values
authentication-Preemptive	When credentials are available and this property is set to the default of false, this property allows Unwired 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 Unwired 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

Proxy Properties

(Applies only to Online Data Proxy) Proxy properties identify the application endpoint and the pool size.

Name	Description	Supported values
User	Not currently used	
Certificate Alias	Not currently used	
Address	Corresponds to the Application end-point provided at the time of registering an application.	Must be a valid application end-point.
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.
Password	Not currently used	

Note: When the application end-point for a registered application is modified under the **Applications** node, you have to manually update the **Address** in the proxy properties of the connection pool.

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 Unwired Platform administration and user roles and passwords required for Sybase Control Center administrator login. See *Enabling Authentication and RBAC for Administrator Logins*.

Task

Perform steps to appropriately configure domain security settings.

Choosing a Security Configuration

Select a security configuration that designates authentication and authorization security providers for the packages in the domain. 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 **Security Configurations** 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 remove a security configuration, check the box adjacent to the configuration name and click **Unassign**. If a security configuration is mapped to one or more MBO packages, it can not be removed.

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

Configure role mapping to authorize client requests to access MBOs and operations. For each security configuration, platform and domain administrators can manage logical role mappings at the package level or at a domain level. Use the corresponding domain or package node in the left navigation pane to configure role mappings accordingly.

Set an appropriate mapping state for each logical role. The state you choose allows you to disable logical roles, allow logical roles to be automatically mapped, or manually define which logical roles are mapped to one or more physical roles. The states of AUTO or NONE require the least administration.

If a developer has defined a logical role, mapping is not required; the logical role is matched to the physical role of the same name and is therefore automatically mapped.

Note: Changes to domain-level role mapping are applied to all domains that share the same security configuration. Likewise, changes to package-level role mapping apply to all instances of the affected package that use the same security configuration, even if the package is deployed in multiple domains.

Setting the Mapping State

Map roles for a package by setting the mapping state. Mapping behavior is determined by the state that exists for the logical role. You can select AUTO or NONE; a third state, MAPPED, is set automatically after you manually map a physical role to the selected logical role.

You can set the mapping state either when managing roles, or earlier, during package deployment. If your logical roles for a package do not automatically match the role names registered in the back-end security system, map corresponding logical and physical names to ensure that users can be authorized correctly.

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. Change the mapping for a logical role, if required:

    - To change the state to either NONE or AUTO, click the list adjacent to the logical role and click the appropriate option.
    - To change the role mapping itself, click the drop-down list adjacent to the logical role and choose **Map Role**. This command displays the Role Mappings dialog that allows you to manually set the physical role mappings. The Role Mappings dialog displays the name of the logical role you are mapping in the text area of the dialog. Once saved, the state automatically changes to MAPPED.

    Note: If the list of available roles is too long, you can search or sort the list to make the view more manageable. See *Mapping a Physical Role Manually* .

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. You can also set the role mapping for each security configuration at the domain level. This allows the role mapping to be shared across packages for the common logical roles. Changing role mapping at the domain level will result in role mapping changes in other domains where the same security configuration is referenced.

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

Unwired 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.

Security Configurations

Sybase Unwired 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 Sybase Control Center .

Security configurations aggregate various security mechanisms for protecting Unwired 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 Sybase Unwired 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 Sybase 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, Unwired Server tries to authenticate and authorize the user, by checking the security repository for:

- Security access policies on the requested resource
- Role memberships

Creating a Security Configuration

Create and name a set of security providers and physical security roles to protect Unwired 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 Sybase Control Center, select **Security**.
2. In the right administration pane, 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.

   The **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*. To configure this value:

   a) Set the cache timeout value in seconds. The default is 3600. To force re-authentication, change this value to 0.

   The **Maximum allowed authentication failure** determines the maximum number of login attempts after which the user is locked. To configure this value:

a) Set the maximum count for authentication failure.

The **Authentication lock duration** determines how long the user is locked after the maximum login attempts is reached . To configure this value:

a) Set the authentication lock duration in seconds.

6. Click Save.
7. Select the tab corresponding to the type of security provider you want to configure: Authentication, Authorization, 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.
10. Select the **General** tab, and click **Validate** to confirm that Unwired Server accepts the new security configuration.
    A message indicating the success of the validation appears above the menu bar.
11. Click **Apply** to save changes to the security configuration, and apply them across Unwired Server.
    A message indicating the success of the application appears above the menu bar.

Security Providers

Different security providers give Unwired Server security features that include authentication, and authorization capabilities.

Configure security providers for Unwired Server by logging in to the server in Sybase Control Center and clicking **Security > Configuration**. Configuring these providers writes changes to the Unwired Server configuration properties file.

For third-party providers, save related JAR files or DLLs in the `<UnwiredPlatform_InstallDir>\UnwiredPlatform\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 Unwired Server is distinct from authorization. You must have at least one authentication module configured in a production deployment of Unwired Server. You can stack multiple providers so users are authenticated in a particular sequence.
- Authorization modules – check the access privileges for an authenticated identity. Sybase recommends that you have at least one authorization module configured in a production deployment of Unwired Server.

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 authenticaton 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

(Not applicable to Online Data Proxy) Optionally, implement multiple login modules to provide a security solution that meets complex security requirements. Sybase 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 SAPSSOTokenLoginModule 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 Unwired Server. In this case, you can also stack another login module so this DCN user can login.

1. Use Sybase 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:

Provider	Authentication Status							
LDAP	pass	pass	pass	pass	fail	fail	fail	fail
NT Login	pass	fail	fail	fail	pass	fail	fail	fail
SSO Token	*	pass	pass	fail	*	pass	pass	fail
Certificate	*	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 Sybase implementation uses the same controlFlag values and definitions as those defined in the JAAS specification.

If you stack multiple providers, you must set the controlFlag attribute for each enabled provider.

Control Flag Value	Description
(Default) required	The LoginModule is required. Authentication proceeds down the LoginModule list.
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	The LoginModule is not required. Irrespective 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 username/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 also succeeds, Unwired Platform picks up roles based on the Windows groups they are in.

Reordering Configured Providers

List stacked security providers for a security configuration to identify them as primary or auxiliary providers. Authentication or authorization by provider take place in the order in which the providers are listed.

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, 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. Click **Save**.
6. 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 whether or not they support authentication or authorization.

Platform administrators can configure application security properties in the Sybase Control Center for Unwired Platform console. These properties are then transcribed to an XML file in the `<UnwiredPlatform_InstallDir>\Servers\UnwiredServer\Repository\CSI\` directory. A new section is created for each provider you add.

LDAP Configuration Properties

(Not applicable to Online Data Proxy) Use these properties to configure the LDAP provider used to authenticate SCC administration logins or to configure the LDAP provider used to authenticate device application logins. If you are creating a provider for device application logins, then Unwired Platform administrators use Sybase Control Center to write these properties to the `<UnwiredPlatform_InstallDir>\Servers\UnwiredServer\Repository\CSI\default.xml` file.

Unwired Server implements a Java LDAP provider through a common security interface used by other Sybase products like Sybase Control Center.

The Java LDAP provider consists of three provider modules, each of which is in the `com.sybase.security.ldap` Java package. This is why the syntax used between Sybase Control Center provider and Unwired Server varies.

- **LDAPLoginModule**– provides authentication services. Through appropriate configuration, you can enable certificate authentication in **LDAPLoginModule**.
- (Optional)**LDAPAuthorizer** or **RoleCheckAuthorizer** – provide authorization services for **LDAPLoginModule**. **LDAPLoginModule** works with either authorizer. In most production deployments, you must always configure your own authorizer. However, if you are authenticating against a service other than LDAP, but want to perform authorization against LDAP, you can use the **LDAPAuthorizer**.

  The RoleCheckAuthorizer is used with every security configuration but does not appear in Sybase 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**.

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 Sybase Control Center, note that properties and values can vary, depending on which module or server type you configure.

Property	Default Value	Description
ServerType	None	Optional. The type of LDAP server you are connecting to: • `sunone5` -- SunOne 5.x OR iPlanet 5.x • `msad2k` -- Microsoft ActiveDirectory, Windows 2000 • `nsds4` -- Netscape Directory Server 4.x • `openldap` -- OpenLDAP Directory Server 2.x The value you choose establishes default values for these other authentication properties: • RoleFilter • UserRoleMembership • RoleMemberAttributes • AuthenticationFilter • DigestMD5Authentication • UseUserAccountControl
ProviderURL	`ldap://localhost:389`	The URL used to connect to the LDAP server. Without this URL configured, Unwired 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>`

Property	Default Value	Description
DefaultSearchBase	None	The LDAP search base that is used if no other search base is specified for authentication, roles, attribution and self registration: 1. `dc=<domainname>,dc=<tld>` For example, a machine in sybase.com domain would have a search base of dc=sybase,dc=com. 2. `o=<company name>,c=<country code>` For example, this might be o=Sybase,c=us for a machine within the Sybase organization.
SecurityProtocol	None	The protocol to be used when connecting to the LDAP server. To use an encrypted protocol, use "ssl" instead of "ldaps" in the url. **Note:** ActiveDirectory requires the SSL protocol when setting the value for the password attribute. This occurs when creating a user or updating the password of an existing user. **Note:** ActiveDirectory requires the SSL protocol when setting the value for the password attribute. This occurs when creating a user or updating the password of an existing user.
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.

Property	Default Value	Description
AuthenticationFilter	For most LDAP servers: `(&amp;(uid={uid})(objectclass=person))` or For Active Directory email lookups: `(&amp;(userPrincipalName={uid})(objectclass=user))[ActiveDirectory]` For Active Directory Windows username lookups: `(&amp;(sAMAccountName={uid})(objectclass=user))` **Note:** Please note these restrictions when using this property to authenticate Sybase Control Center administration use cases only: • Do not use special characters (for example, `, = : ' " * ? &`) in user names identified with this property. • Do not use Chinese or Japanese characters in the user name	The filter to use when looking up the user. When performing a username based lookup, this filter is used to determine the LDAP entry that matches the supplied username. The string "`{uid}`" in the filter is replaced with the supplied username.

Property	Default Value	Description
	or passwords of this property.	
AuthenticationScope	onelevel	The authentication search scope. The supported values for this are: • `onelevel` • `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.
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. However, the LDAP attributer may also use this DN to create the users in the LDAP server. When the self-registration feature is used, this user may also need the requisite permissions to create a user record. This behavior can occur if you do not set useUserCredentialsToBind to `true`. In this case, the LDAP attributer uses this DN to update the user attributes.

Property	Default Value	Description

BindPassword	none	BindPassword is the password for BindDN, which is used to authenticate any user. BindDN and BindPassword are used to separate the LDAP connection into units.
		The AuthenticationMethod property determines the bind method used for this initial connection.
		Sybase recommends encrypting passwords and provides a password encryption utility for the purpose. If you encrypt BindPassword, include `encrypted=true` in the line that sets the option. For example:
		`<options name="BindPassword" encrypted="true" value="1snjikfwregfqr43hu5io..."/>`
		If you do not encrypt BindPassword, the option might look like this:
		`<options name="BindPassword" value="s3cr3T"/>`

RoleSearchBase	none	The search base used to retrieve lists of roles. If this property is not configured, the value for DefaultSearchBase is used.

Property	Default Value	Description
RoleFilter	For SunONE/iPlanet: `(&amp;(objectclass=ldapsubentry) (objectclass=nsroledefinition))` For Netscape Directory Server: `(|(objectclass=groupofnames) (objectclass=groupofuniquenames))` For ActiveDirectory: `(|(objectclass=groupofnames) (objectclass=group))`	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.
RoleMemberAttributes	For Netscape Directory Server and OpenLDAP Server: member,uniquemember	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 only has a default value when the Netscape server type is chosen.
RoleNameAttribute	cn	The attribute of the role entry used as the role name in Unwired Platform. This is the role name displayed in the role list or granted to the authenticated user.
RoleScope	onelevel	The role search scope. The supported values for this are: • `onelevel` • `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.
UserRoleMembershipAttributes	For iPlanet/SunONE: nsRoleDN For ActiveDirectory: memberOf For all others: none	The user's role membership attributes property is used to define an attribute that a user has that contains the DN's of all of the roles as user is a member of. These comma-delimited values are then cross-referenced with the roles retrieved in the role search base and search filter to come up with a list of user's roles. **Note:** If SkipRoleSearch property is set to true, then these comma-delimited values will not be cross-referenced with the roles retrieved in the role search base and role search filter. See *Skipping LDAP Role Lookups (SkipRoleLookup)*. **Note:** If you use nested groups with ActiveDirectory, you must set this property to "tokenGroups". See *Using LDAP Nested Groups and Roles*.
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".
Referral	ignore	The behavior when a referral is encountered. The valid values are those dictated by LdapContext, for example, "follow", "ignore", "throw".
DigestMD5Authentication-Format	DN For OpenLDAP: User-name	The DIGEST-MD5 bind authentication identity format.

Property	Default Value	Description
UseUserAccountControlAttribute	For ActiveDirectory: true	When this property is set to true, the UserAccountControl attribute is used for detecting disabled user accounts, account expirations, password expirations and so on. ActiveDirectory also uses this attribute to store the above 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. **Note:** For more information, see *controlFlag Attribute Values*. **Note:** controlFlag is a generic login module option rather than an LDAP configuration property.
EnableLDAPConnectionTrace	None	Enables LDAP connection tracing. The output is logged to a file in temp directory. The location of the file is logged to the server log.

Configuring LDAP 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 the keytool utility (**keytool.exe**) to import the certificate into the truststore.

1. Run the following console command: **keytool.exe -import -keystore <UnwiredPlatform_InstallDir>\Servers\UnwiredServer\Repository\Security\truststore.jks -file <your cert file and path> -alias ldapcert -storepass changeit**.
2. Restart Sybase Unwired Platform services.
3. Log in to Sybase Control Center for Sybase Unwired Platform.
4. In the navigation pane of Sybase Control Center, expand the Security folder and select **admin**.
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. See *LDAP Configuration Properties* and *LDAP Login and Authorization Modules*.
   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**.

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 Sybase Control Center.

Table 5. Authentication properties

Properties	Default Value	Description
Extract Domain From Username	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.
useFirstPass	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.
tryFirstPass	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.
clearPass	false	If set to true, the login module clears the user name and password in the shared context when calling either commit or abort.

Properties	Default Value	Description
storePass	false	If set to true, the login module stores the user name and password in the shared context after successfully authenticating.

NoSec Configuration Properties

A NoSec provider offers pass-through security for Unwired Server. In development environments, you can apply a NoSec security provider for authentication and authorization modules. However, never use 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

However, you need to configure only authentication properties for a NoSec provider.

Table 6. 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.
useFirstPass	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.

Property	Default Value	Description
tryFirstPass	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.
clearPass	false	If set to true, the login module clears the user name and password in the shared context when calling either commit or abort.
storePass	false	If set to true, the login module stores the user name and password in the shared context after successfully authenticating.

Certificate Authentication Properties

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 7. CertificateAuthenticationLoginModule properties

Property	Description
Implementation class	The fully qualified class that implements the login module. `com.sybase.security.core.CertificateAuthenticationLoginModule` is the default class.
Provider type	`LoginModule` is the only supported value.
Control flag	Determines how success or failure of this module affects the overall authentication decision. `optional` 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.

Property	Description
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 muse enable OCSP in Unwired Server. This provider uses the values defined as part of the SSL security profile. Revoked certificates result in authentication failure when both of these conditions are met: • revocation checking is enabled • 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.
Trusted certificate store	(Optional) The file containing the trusted CA certificates (import the issuer certificate into this certificate store). Use this property and `Store Password` property to keep the module out of the system trust store. The default Unwired Server system trust store is `<UnwiredPlatform_InstallDir\Servers\UnwiredServer\Repository\Securitytruststore\truststore.jks`. If you do not specify a store location:: • Unwired Server checks to see if a store used by the JVM (that is, the one defined by the `javax.net.ssl.trustStoreType` system property. • If the system property is not defined, then this value is used: `${java.home}/lib/security/jssecacerts` • If that location also doesn't exist, then this value is used: `${java.home}/lib/security/cacerts` **Note:** This property is required only if Validate certificate path is set to true.

Property	Description
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 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 `javax.net.ssl.trustStorePassword` property.
Trusted certificate store provider	(Optional) The keystore provider. For example, "SunJCE." **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 `javax.net.ssl.trustStoreProvider` 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.
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 `javax.net.ssl.trustStore` 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.

Property	Description
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. For information about adding a certificate to the truststore, see *Using Keytool to Generate Self-Signed Certificates and Keys* in *Security*.

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 8. CertificateValidationLoginModule properties

Property	Description
Implementation class	The fully qualified class that implements the login module. `com.sybase.security.core.CertificateValidationLoginModule` 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:
	`crl.1.uri=http://crl.verisign.com/ThawtePersonalFreemailIssuingCA.crl`
	`crl.2.uri=http://crl-server/`

Provider type	`LoginModule` is the only supported value.

Property	Description
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 `false`. If you are implementing this provider with a DCN security configuration, and it's also not used with SSO, then set this property to `true`.`False` 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 muse enable OCSP in Unwired Server. This provider uses the values defined as part of the SSL security profile. Revoked certificates result in authentication failure when both of these conditions are met: • revocation checking is enabled • OCSP properties are configured correctly
Trusted certificate store	(Optional) The file containing the trusted CA certificates (import the issuer certificate into this certificate store). Use this property and `Store Password` property to keep the module out of the system trust store. The default Unwired Server system trust store is `<UnwiredPlatform_InstallDir\Servers\UnwiredServer\Repository\Securitytruststore\truststore.jks`. If you do not specify a store location:: • Unwired Server checks to see if a store used by the JVM (that is, the one defined by the `javax.net.ssl.trustStoreType` system property. • If the system property is not defined, then this value is used: `${java.home}/lib/security/jssecacerts` • If that location also doesn't exist, then this value is used: `${java.home}/lib/security/cacerts` **Note:** This property is required only if Validate certificate path is set to true.

Property	Description
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 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 `javax.net.ssl.trustStorePassword` property.
Trusted certificate store provider	(Optional) The keystore provider. For example, "SunJCE." **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 `javax.net.ssl.trustStoreProvider` 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.
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 `javax.net.ssl.trustStore` 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.

Property	Description
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. For information about adding a certificate to the truststore, see *Using Keytool to Generate Self-Signed Certificates and Keys* in *Security*.

SAP SSO Token Authentication Properties

The SAPSSOTokenLoginModule has been deprecated, Use the HttpAuthenticationLoginModule when SAP SSO2 token authentication is required. This authentication module will be removed in a future release.

Table 9. SAPSSOTokenLoginModule properties

Property	Description
Implementation class	(Required) – the fully qualified class that implements the login module. `com.sybase.security.sap.SAPSSOTokenLoginModule` is the default class.
Provider type	(Required and read-only) – `LoginModule` is the only supported value.
Control flag	(Required) – `optional` is the default value. Determines how success or failure of this module affects the overall authentication decision.

Property	Description
SAP server URL	(Required) – the SAP server URL that provides the SSO2 token. This may or may not be the same server that authenticates the user. If providing and authenticating servers are different, you must import the SAP Token provider server certificate or one of its CA signers into the Unwired Server truststore in addition to that of the authenticating server to enable HTTPS communication. In environments where the servers are different, the basic flow is: **1.** Unwired Server passes credentials over HTTPS to the token granting service. **2.** An SSO2Token cookie is returned to Unwired Server. **3.** The SSO2Token flows to the authenticating server, which could be an SAP EIS or a server that hosts a Web service bound to SAP function modules. **Note:** The SAP Server URL must be configured to require BASIC authentication, not just FORM based authentication.
Clear password	(Optional) – if set to `True`, the login module clears the username and password in the shared context.
Disable server certificate validation	(Optional) – the default is False. If set to `True`, disables certificate validation when establishing an HTTPS connection to the SAP server using the configured URL. Set to `True` only for configuration debugging.
SAP server certificate	(Optional) – name of the file containing the SAP certificate's public key in `.pse` format. This is required only when token caching is enabled by setting a SAP SSO token persistence data store value.
SAP server certificate password	(Optional) – password used to access the SAP server certificate.

Property	Description
SAP SSO token persistence data store	(Optional) – JNDI name used to look-up the data source to persist the retrieved SSO2 tokens. Set to "jdbc/default" to store tokens in the Unwired Server CDB. If unconfigured, some caching is still done based on the "Authentication cache timeout interval" property associated with the security configuration setting. If you use the default setting, you do not need to set SAP SSO token persistence data store, SAP server certificate, SAP server certificate password, or Token expiration interval properties. To enable token caching through the SAPSSOTokenLoginModule: **1.** Set the SAP SSO token persistence data store value to "jdbc/default." **2.** Download and install the SAP SSO2 token files. See *Installing the SAP SSO2Token Files on Unwired Server Hosts* in the *Security* guide. **3.** Specify the correct value for the SAP server certificate, SAP server certificate, SAP server certificate password and Token expiration interval properties.
Store password	(Optional) – if set to true, the login module stores the username/password in the shared context after successfully authenticating the user.
Token expiration interval	(Optional) – this property is ignored when the SAP SSO token persistence data store property is not configured. It specifies the token validity period, after which time a new token is retrieved from the SAP EIS. The default value is 120 seconds. Keep in mind that: • The "Token expiration interval" cannot exceed the "Token validity period", which is the amount of time defined in the back-end SAP server for which the token is valid. • The "Authentication cache timeout" property must be less than the "Token expiration interval" property value.

Property	Description
Try first password	(Optional) – if set to `True`, the login module attempts to retrieve the username/password from the shared context, before calling the callback handler.
Use first password	(Optional) – if set to `True`, the login module attempts to retrieve the username/password only from the shared context, and never calls the callback handler.
HTTP connection timeout interval	The value, in seconds, after which an HTTP(s) connection request to the EIS times out. If the HTTP connection made in this module (for either user authentication or configuration validation) does not have a time out set, and attempts to connect to an EIS that is unresponsive, the connection hangs, which could potentially cause Unwired Server to hang. Setting the timeout interval ensures authentication failure is reported without waiting for ever for the server to respond.

Preconfigured User Authentication Properties

The PreConfiguredUserLoginModule authenticates the Unwired Platform Administrator user whose credentials are specified during installations.

This login module is recommended only to give the Platform administrator access to Sybase 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 username/password against the configured user. Upon successful authentication, the configured roles are added as Principals to the Subject.

Table 10. PreConfiguredUserLoginModule properties

Property	Description
User name	A valid user name. Do not use any of these restricted special characters: `, = : ' " * ? &`.
Password	The encoded password hash value.

Property	Description
Roles	Comma separated list of roles granted to the authenticated user for role-based authorization. Platform roles include "SUP Administrator" and "SUP Domain Administrator". 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. **Note:** If you use other values, ensure you map Unwired Platform roles to the one you define here.

HTTP Basic Authentication Properties

The HttpAuthenticationLoginModule provider authenticates the user with given credentials (user name and password) against the secured Web server (SWS) 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) or network edge authentication.

This provider can be configured for authenticating the user when:

- using only the specified username/password
- using only the specified client value(s)
- first attempting token authentication and if it fails, reverting to basic authentication using the supplied username/password. This could be helpful when using the same security configuration for authenticating users with a token, such as device users hitting network edge, and when DCN requests from within a firewall present only the username/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 11. HttpAuthenticationLoginModule configuration options

Configuration Option	Default Value	Description
URL	None	The HTTP(S) URL that authenticates the user. For single sign-on, this is the server URL from which Unwired Server acquires the SSO cookie/token.

Configuration Option	Default Value	Description
Disable certificate validation	False	(Optional) The default is false. If set to true , disables 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) A name of the cookie that is set in the session between the LoginModule and the SWS and holds the SSO token for single sign-on. The provider looks for this cookie in the connection to the SWS. If found, it is added to the authenticated subject as a named credential. The authentication provider ignores the status code when a SSO cookie is found in the session. If the cookie is found, 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.
Successful connection status code	200	HTTP status code interpreted as success when connection is established to the SWS.

Configuration Option	Default Value	Description
HTTP connection timeout interval	1 minute	The value, in seconds, after which an HTTP(s) 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 time out set, and attempts to connect to a Web-based authentication service that is unresponsive, the connection hangs, which could potentially cause Unwired Server to hang. Setting the timeout interval ensures authentication failure is reported without waiting for ever for the server to respond.
SendClientHttpValuesAs	None	Comma separated list of strings that indicate how the ClientHttpValuesToSend should be sent to the HTTP server. For example: `SendClientHttpValuesAs=header:`*`header_name`*`, cookie: `*`cookie_name`* **Note:** If the user should be authenticated only using the supplied username/password, then this property does not apply.

Configuration Option	Default Value	Description
ClientHttpValuesToSend		A comma separated list of client HTTP values that should be sent to the HTTP server. For example: `ClientHttpValuesToSend=`*`client_personalization_key, client_cookie_name`* This property should be set if token authentication is used. Setting the property "ClientHttpValuesToSend" triggers token authentication. Unless TryBasicAuthIfTokenAuthFails is configured to true in conjunction with ClientHttpValuesToSend, only token authentication will be attempted. **Note:** If the user should be authenticated only using the supplied username/password, then this property does not apply.
SendPasswordAsCookie	None	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/cookies to the SWS without impacting the password field.

Configuration Option	Default Value	Description
TryBasicAuthIfTokenAuthFails	False	Option that specifies if the provider should attempt basic authentication using the specified username/password credentials if token authentication is configured and it fails. This property is applicable only if token authentication is enabled. **Note:** If the user should be authenticated only using the supplied username/password, then this property does not apply.
UsernameHttpHeader	None	Http response header name that is sent back by the HTTP server with the username retrieved from the token. The retrieved username is added as a SecNamePrincipal upon successful authentication. **Note:** If the user should be authenticated only using the supplied username/password, then this property does not apply.

Configuration Option	Default Value	Description
regexForUsernameMatch	None	Regular expression to use for matching the supplied username with the username returned by the HTTP server in the UsernameHttpHeader. The string "{username}" in the regex is replaced with the specified username before using it. If specified, it is used to match the username retrieved from the UsernameHttpHeader to the username specified in the callback handler. It they do not match, it results in authentication failure. If they match, both the specified username and the retrieved username are added as SecNamePrincipals to the authenticated subject. **Note:** If the user should be authenticated only using the supplied username/password, then this property does not apply.

Configuration Option	Default Value	Description
TokenExpirationTimeHttpHeader	None	HTTP response header name that is sent back by the HTTP server with the validity period of the token in milliseconds from the start of January 1, 1970. If the header is returned in the HTTP response from the SWS, 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. **Note:** If the user should be authenticated only using the supplied username/password, then this property does not apply.

Configuration Option	Default Value	Description
TokenExpirationInterval	0	Property to specify 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:** If the TokenExpirationTimeHttpHeader value returned by the SWS is less than the value configured for the TokenExpirationInterval property, it results in authentication failure. **Note:** If the user should be authenticated only using the supplied username/password, then this property does not apply.
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.

Auditor Filter Properties Reference

(Not applicable to Online Data Proxy) Configure multiple resource classes when defining an auditor for a named security configuration.

Filter resource classes require a specific syntax. Based on that syntax, an audit token is supplied to the core CSI classes. This audit token identifies the source for core audit requests of operations, such as auditing the results for authorization decisions, 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. CSI core will able to audit a large number of items.

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.
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. **Note:** that this is not a provider-generated audit record. CSI core will generate this audit record automatically after receiving the provider's decision. The resource ID is not used.	• Provider identifier • Decision (yes, 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 Subject identifier if authentication successful.	• Decision (yes or no) • Context ID

Resource Class	Action	Description	Attributes
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
subject	authorization.role	The result of a resource-based authorization request. The resource ID is the subject ID.	• Resource name • The access requested Decision (yes, no or abstain) • Supplied subject identifier (if different from context subject) • Context ID
subject	authorization.resource	The aggregate decision authorization decision after considering each of the appropriate provider's authorization decision. The resource ID is the subject ID.	• Resource name • Access requested Decision (yes, 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

Resource Class	Action	Description	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, no) • Subject ID supplied (if different from context subject) • Context ID

Examples

- **Example 1 –** enable auditing of all of the CSI core resource classes that involve a deny decision:

  ```
  (ResourceClass=core.*,Decision=Deny)
  ```
- **Example 2 –** enable auditing for all core resource classes where the action is the subject modification action:

  ```
  Resource=core.*,Action=subject.modify.*)
  ```

Roles and Mappings

Role mapping occurs when an administrator maps logical roles to physical roles using Sybase Control Center as part of a security configuration or a deployment package. The physical roles are the roles and groups in the underlying security repository. The mapped role determines the security role requirement for a user at runtime to access a resource that is using the security configuration on which the mapping is defined.

In Unwired Platform, the mapped role determines what security roles apply to users when they attempt to perform an operation from the mobile application (device users) or Sybase Control Center (administrators).

Role mappings are defined as part of a security configuration that you can assign to a particular domain. Administrators can assign the same security configuration to multiple domains; ensure that these mappings are suitable for all domains to which the security configuration is assigned. Consider an example where security configuration is shared between domainA and domainB.

1. The platform administrator (the administrator assigned the SUP administration role) creates a security configuration called AllDomains.
2. The platform administrator assigns the AllDomain to the domain, and maps the EmpRole role to SalesGroupRole in the security repository used by that configuration.

This change that is specific to just domainA is also implemented in domainB even though the domain administrator of domainB did not explicitly make, or require, the change. But the role mapping is propagated to domainB as well. To avoid this, the Unwired Platform administrator may want to create multiple security configurations so that underlying mechanisms can stay the same, but specific role mappings can be made for each.

For device user security, there is an increased flexibility for packages as they are deployed. If a security configuration is inappropriate, or if a role is not mapped at all that is used by the package, the platform or domain administrator can override or extend the role mappings defined for the security configuration. Package-level role mappings always take precedence in such a scenario.

Applications

An Application is the runtime entity that can be directly correlated to a native or mobile workflow application. The application definition on the server establishes the relationship among packages used in the application, domain that the application is deployed to, user activation method for the application, and other application specific settings.

- For native replication/messaging applications, one or more MBO packages can be assigned to an application. If the application developer uses the same package in a different application, the MBO package must be assigned to that application.
- For the mobile workflow applications, all MBO packages are accessible in all domains, so no MBO packages need to be assigned to the mobile workflow application if you are using the default (HWC). However, for a customized container with an application ID other than the default mobile workflow application (HWC), the corresponding application's MBO packages must be assigned to the customized mobile workflow application (customized unique application ID).
- For Online Data Proxy applications, no MBO package assignments are needed as well.

Applications are managed and monitored by administrators on the Sybase Control Center. They are created automatically or manually through Sybase Control Center.

An application ID uniquely identifies an application to Unwired Server. Application connection templates enable administrators to manually register application connections in Unwired Server with predefined settings. Templates also enable automatic activation of devices (described later). Users are associated with one or more applications through application connections. Administrators can view application users in Sybase Control Center as soon as a user logs onto the application from a device.

Setting Up Application and User Connections

The basic steps for setting up application and user connections involve creating an application, and registering application connections to associate users to the application.

Application Creation

There are two ways an application gets created - automatic and manual.

Automatic Application Creation

Applications are created automatically, when the system administrator deploys a package. The mobile workflow application (HWC) is created during Unwired Server installation.

An application is created automatically when an MBO package is deployed to the server. In case of upgrade from a previous server version, an application is created for all deployed MBO packages. The default name of an application is the same as the package name.

Note: An application is primarily used for tracking purpose. An application connection template is also created and used for automatic registration for native applications (MBO package client applications).

For Online Data Proxy, there is no such automatic creation of applications. Applications must be created manually.

Manually Creating Applications

Create an application manually by assigning a unique application ID and other key application properties, such as domain, MBO package, security configuration, among others. At this time, the manual process is only needed for Online Data Proxy applications or when using a Hybrid Web Container built using the iOS sample, where developers can use their own application IDs for workflow applications.

Launching the Application Creation Wizard

Use the Application Creation wizard to register an application.

1. In the left navigation pane of Sybase Control Center, click the **Applications** node and select the **Applications** tab in the right administration pane.
2. To register an application, click **New**.
   The Application Creation wizard is displayed.

Setting General Application Properties

Provide general application properties such as the application ID, description, security configuration and domain details while registering the application.

1. In the Application Creation Wizard, enter a unique **Application ID**, following application ID guidelines.
2. Enter a **Display name** and **Description** for the application.
3. Select the appropriate security configuration from the **Security Configuration** drop-down list.
4. Select the appropriate domain from the **Domain** drop-down list.
5. (Optional) Assign one or more packages as desired.
6. Click **Finish** to register the application with the configured settings.

Application ID Overview

Applications can be directly correlated to a native application or mobile workflow container instance on device. A native application is the single binary deployed to device which may use one or more MBO packages. The mobile workflow application is a collection of workflow packages and constitutes as one application. One or more MBO packages can be assigned to an application. If application developers want to use the same package in a different application, they can do that by assigning the MBO package to that application using Sybase Control Center.

An application ID uniquely identifies the application and must be used to register an application connection or the application template for automatic activation of the application. An application ID is also used in the device application for activation of its application connection. Depending on the choice of the activation option, invoke the provided Application APIs to register the application connection.

Modifying Application Properties

Associate the application with one or more domains and packages. (Optional for OData SDK Android and iOS clients) Associate the application with one or more customization resource bundles.

1. In the left navigation pane of Sybase Control Center, click the **Applications** node, and, in the right administration pane, select the **Applications** tab.
2. Select an application, and click **Properties**.
3. Click the **Domains and Packages** tab.
   a) Select the domains to associate with the application.

      To see more domains, click **+**. In Available Domains, select one or more domains from the list.

   a) Select the packages to associate with the application.

      To see more packages, click +. In Available Packages, select one or more packages from the list.

4. (Optional for OData SDK Android and iOS clients) Click the **Customization Resource Bundles** tab.
   a) Select the customization resource bundles to associate with the application.

      To see more customization resource bundles, click +. In Select File to Upload, select a JAR file.
5. Click **OK**.

Customization Resource Bundles

(Applies only to OData SDK, Android and iOS clients) Customization resource bundles enable you to associate deployed client applications with different versions of customization resources.

A customization resource bundle is a JAR file that includes a manifest file of name and version properties. The customization resource bundle does not contain any information that binds or helps bind to applications; it can be deployed, undeployed, or exported during update of an application through Sybase Control Center. A deployed customization resource bundle is read-only.

Implementation task flow:

1. (Client Application Developer) Invokes the OData SDK API that downloads the customization resource bundle, which ties the application to the device. This enables the customization resource bundle to reach the client application. See *Developer Guide: OData SDK*.
2. (Developer) Generates the JAR with the MANIFEST.MF, which includes these required properties:
   - `Customization-Resource-Bundle-Name`
   - `Customization-Resource-Bundle-Version`
3. (System Administrator) Uses Sybase Control Center to upload the customization resource bundle, deploy it to an application, and assign it to an application connection or application connection template.
4. (Mobile Device) Deployed client applications are directed to the appropriate version of the application.

Customization Resource Bundle Recommendations

There are a variety of recommendations for working with customization resource bundles.

- You can use customization resource bundles only for OData SDK clients (Android and iOS).
- 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`
  - The property values cannot include a colon (":").
  - Sybase recommends the file size not exceed 5MB. File size is not enforced, but the larger the file, the slower the performance, subject to device platform hardware capabilities.

  See *Managing Customization Resource Bundles* in *Developer Guide: Unwired Server Management API* for information about the administration API that allows programmatic access to this functionality.

- You can deploy the same customization resource bundle to different applications, and it is treated independently for each application. The primary key is: application ID, customization resource bundle name, and version.
- Each deployed 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 not assigned, by default, 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 deployed, meaning the format does not change.
- You can deploy 0, 1 or more customization resource bundles to one application as long as the name and version combination is unique.
- For each application connection or application connection template, you can assign only one primary customization resource bundle. Any deployed customization resource bundle is accessible to any application connection, regardless if assigned to the application connection properties or application connection template. This knowledge about the bundle name version allows any authenticated device/application user to access any customization resource bundle stored in the server.
- 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.

Deploying Customization Resource Bundles

Add a customization resource bundle to an application, and optionally assign it to its application connection, and application connection template. At any given time, only one customization resource bundle can be assigned to an application connection or application connection template. The most recent assignment replaces a previous assignment.

You must log in as an SUP Platform Administrator.

1. In the left navigation pane, click the cluster-level **Applications** node, and, in the right administration pane, select the **Applications** tab.
2. Select one application, and click **Properties**. In the dialog, 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 **OK**. 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 deployed bundle to application connections or application connection templates with the same application ID. If no check box is selected, there is no automatic assignment.

   Note: You can make these assignments at a later time.
5. (Optional) Add another customization resource bundle to the application.
6. Click **OK**.

Assigning a 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 left navigation pane, click the cluster-level **Applications** node, and, in the right administration pane, select the **Applications** tab.
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 a 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 left navigation pane, click the cluster-level **Applications** node, and, in the right administration pane, select the **Applications** tab.
2. Select one application, and click **Properties**. In the dialog, 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 Unassignbuttons.
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 Customization Resource Bundles

Unassign a customization resource bundle from an application.

1. In the left navigation pane, click the cluster-level **Applications** node, and, in the right administration pane, select the **Applications** tab.
2. Select one application, and click **Properties**. In the dialog, select the **Customization Resource Bundles** tab.
3. (Optional) Click **Refresh** to update the list of deployed customization resource bundles for this application.
4. Click **Unassign** to launch the dialog. The list of unassignable application connections and application connection templates appears in the respective tabs.

   Click **Yes** to confirm the unassignment for the customization resource bundle.

Managing Customization Resource Bundles

(Applies only to OData SDK, Android and iOS clients) Use Sybase Control Center to view deployed customization resource bundles, and to export and delete customization resource bundles.

Viewing Deployed Customization Resource Bundles

View the customization resource bundles deployed to an application.

1. In the left navigation pane, click the cluster-level **Applications** node, and, in the right administration pane, select the **Applications** tab.
2. Select one application, and click **Properties**. In the dialog, 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 Customization Resource Bundles

Export a customization resource bundle from an application.

1. In the left navigation pane, click the cluster-level **Applications** node, and, in the right administration pane, select the **Applications** tab.
2. Select one application, and click **Properties**. In the dialog, 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 file location and click **OK** to create a customization resource bundle JAR file.
7. (Optional) Export another customization resource bundle JAR file for the application.

Deleting a 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 left navigation pane, click the cluster-level **Applications** node, and, in the right administration pane, select the **Applications** tab.
2. Select one application, and click **Properties**. In the dialog, 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.

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 applictions 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 Sybase Control Center, click the **Applications** node and select the **Applications** tab in the right administration pane.
2. Select the application and click **Delete**.

Application Connection Activation Options

An application connection can be activated from the device by either providing valid credentials, or an activation code for a pre-registered application connection. The credentials-based approach, referred to from this point on as automatic registration, relies on application connection template properties (application ID, security configuration, automatic registration enabled). The activation-based approach, referred to from this point as manual registration, relies on matching user name and activation code sent from the device to an existing application connection registered for the user.

Information and guidelines:

- Application templates are used for automatic activation. Therefore, when setting up the application template for automatic registration, be sure to set up the security configuration, domain, the application ID, and automatic registration enabled properties in application settings. Those are used for automatic application registration.

  When a client application connects to the server with its application ID and credentials, and requests automatic registration, the 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),the security configuration in the template is used to validate the credentials. Upon successful validation of those credentials, the user identity is registered in the Unwired Server. The client application may also include the security configuration as part of the username and in that case, the security configuration (in addition to application ID) is used to look up a matching template. If no or multiple templates are detected, the registration request fails. For details on how user names and security configuration names are processed when an email address is used, see *Considerations for Email Addresses as Username* in the *Security* guide.
- Supported device client activation options:

Device Client Type	Automatic Registration	Manual Registration
Workflow	X	X
Native	X	X
Online Data Proxy	X	X

Managing and Searching for Applications

You can view the applications registered through the Sybase Control Center

1. In the left navigation pane, click the **Applications** node.
2. In the right administration pane, click the **Application** tab.
   You can view the list of registered applications.

Viewing Assigned Connections

View the properties of the connections assigned to an application.

1. In the right administration pane, select the **Applications** tab.
2. Select the application from the list.
3. Click **Application Connections**.
4. Click **Refresh** to refresh the list that displays the application connections.

Viewing Assigned Application Users

View the list of the users assigned to an application

1. In the right administration pane, select the **Applications** tab.
2. Select the application from the list.
3. Click **Application Users**.
4. Click **Refresh** to refresh the list that displays the application users.

Viewing Correlated Application Details

Select one or more applications, then view correlated application details in several categories, including packages, application users, and application connections.

1. In the left navigation pane, click the **Applications** node.
2. In the right administration pane, click the **Application** tab.
   You can view the list of registered applications.
3. Select one or more applications from the list.
4. Select on of the buttons to view related packages, application users, or application connections.

- **Application Users** – view correlated applications.
- **Application Connections** – view correlated applications.

5. In Review Assignment, check the information.
6. Click **OK**.

Refreshing the Application View

Refresh the list of all available applications registered through the Sybase Control Center.

1. In the right administration pane, select the **Applications** tab.
2. To view the list of registered applications, click **Refresh**.

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.

Searching from the Default View

Search for applications that are registered in the Sybase Control Center.

1. In the right administrations pane, select the **Applications** node.
2. To set the search criteria, select the criteria from the **Search** drop-down list.
3. Add a search string.
4. Click **Go**.
   All the applications that match the search criteria provided are populated in the table.

Application Users

In Unwired Platform, an application user is an identity registered as the user of one or more versions of a device application. An application user can have multiple devices.

When a user is on-boarded the identity becomes visible and manageable in Sybase Control Center. The process varies, depending on the application type, the registration type, and the version of the client runtime:

- For native applications using the current version of client runtimes, an application user is automatically registered when the application connection is successfully registered.
- For native applications using the current version of client runtimes but are manually registered, an application user is registered at application connection registration time.
- For native applications using previous version of client runtimes, an application user is automatically registered upon first successful authentication with the package.

- For native messaging applications, an application user is automatically registered upon first successful synchronization after manual registration.
- For Workflow and Online Data proxy applications using previous or current client runtimes, the application user is registered upon successful registration, whether manual or automatic.

Once registered and on-boarded, the platform administrator can view the user name and the security configuration that was used to authenticate the user. In addition, the administrator can remove users that no longer exist.

Note: SAP DOE-C package users are not registered in Unwired Server. Those users are authenticated by their respective DOE back-end servers.

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 Sybase 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.

Checking Application User Assignments

Check which applications are used by registered users.

1. In the left navigation pane of Sybase 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 right administration pane, select the **Applications Users** node.
2. Choose the criteria from the **Search** drop-down list for which you want to search for the required user.
3. 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 right administration pane, select the**Applications** node, then the **Application Users** tab.
2. Click **Refresh** to view the current information of all users.

Application Connections

Application connections associate an application instance with a user. An application may be used by many users, and a user may be associated with many applications.

Application ID Guidelines

Follow these guidelines for choosing an appropriate application ID while registering application connection for use by native MBO, workflow, or Online Data Proxy clients. Failure to specify the correct application ID would result in failure when the client tries to activate itself even though the user name and activation code do match.

Registration Type	Application ID Guidelines
Workflow client	• 2.0.1 or earlier – leave the application ID empty. • 2.1 or later – use preexisting "HWC" template, or, if 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 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 synchronization, set other template properties (such as synchronization-related properties in Connection category) as required. For Android native MBO application, this recommendation applies starting with 2.1.1.

Registration Type	Application ID Guidelines
Online Data Proxy client	Register application connection using the template created for the application. Existing templates can be found in the **Applications > Application Connection Template** tab.

Registering or Reregistering Application Connections

Use Sybase Control Center to trigger the registration of an application connection, or reregister an application connection when the activation code has expired.

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.
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) 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 mobile workflows (containers). Use as is, or customize. If you use the HWC template, Application ID must be set to HWC.
      - Custom - customized templates are listed.
5. Change the default field values for the template you have chosen. If you are using the default template, you must provide the server name.

   If you are using Relay Server, ensure the correct values are used.

   - **Server name**- the DNS name or IP address of the primary Unwired 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.
   - **Port**- the port used for messaging connections between the device and Unwired Server. If using relay server, this is the Relay Server port. Default: 5001.
   - **Farm ID**- a string associated with the relay server farm ID. Can contain only letter A-Z (uppercase or lowercase), numbers 0-9, or a combination of both. Default: 0.

- **Application ID**- the application ID registered for the application. The value differs according to application client type - native application, workflow, or Online Data Proxy client. See *Application ID Overview* for guidelines.
- **Security Configuration**- select the security configuration relevant for the application connection.
- **Domain**- select the domain for which you want to register the application connection with. A domain is not required for registering application connections for workflow 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.
- **Activation code length** - the number of characters in the activation code. If you are reregistering or cloning a device, this value cannot be changed.
- **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**

Searching for Application Connections

Set search criteria to filter connections viewed in the Application Connections tab

1. In the right navigation pane, 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.

Deleting Application Connections

Delete an application connection to remove a user assignment to an application connection.

1. In the left navigation pane of Sybase 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** node.

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.
   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 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** node.
3. Check the box adjacent to the connection you want to clone and click **Clone**.
4. Edit the configuration settings associated with the application connection.
   - **Server name**- the DNS name or IP address of the primary Unwired 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.
   - **Port**- the port used for messaging connections between the device and Unwired Server. If using relay server, this is the relay server port. Default: 5001.
   - **Farm ID**- a string associated with the relay server farm ID. Can contain only letter A-Z (uppercase or lowercase), numbers 0-9, or a combination of both. Default: 0.
   - **Application ID**- the application ID registered for the application.
   - **Security Configuration**- select the security configuration relevant for the application connection.
   - **Domain**- select the domain for which you want to register the application connection with.
   - **Activation expiration**- 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**

Tracing Application Connections

Send a request to Unwired Server 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**.
   The application connection status must be "online" to retrieve the logs.
4. Click **OK**.
5. When the application connection is online, check the application connection log. The default location for single node and cluster installations is `<`*`UnwiredPlatform_InstallDir`*`>\UnwiredPlatform\Servers\UnwiredServer\logs\ClientTrace`.

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**.

Application Connection Templates

An application connection template is a model or pattern used to standardize connection properties and values for a specific application connections so that they can be reused. A template allows you to quickly create actual application connections.

Application Connection Templates are automatically created at the MBO package deployment time with the key properties already set based on the MBO package deployment settings. Commonly used properties include, the application ID, security configuration(used for automatic on-boarding), domain (used to synchronize with the MBO packages of that application).

In addition to those settings, other properties such as those in "Connections" and "Security Settings" inherit default values based on the server configuration. Those values are automatically propagated to the client application and used as the default values. Administrators must ensure that those settings are accurate for a production deployment; if not modify them as required to meet the connection needs of various applications.

Additionally, two built-in templates are also provided if you want 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.

You can use these built-in templates or create new ones as your deployment environment dictates.

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 Settings*:
   - Apple Push Notifications
   - Application Settings
   - BlackBerry Push Notifications
   - Connection
   - Custom Settings
   - Device Advanced
   - Device Info
   - Proxy
   - Security Settings
   - User Registration
7. Click **OK**.

Managing Properties of Application Connection Template

Manage the different categories of properties set for application connection templates.

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 **Properties**.
4. In the Template dialog, select the category you want to edit and modify the property and value.
5. Click **OK**.

Deleting an Application Connection Template

When an application connection template is no longer needed, delete the template. Deleting an application connection template can also be used to prevent automatic activation when a template was used for that purpose. Therefore, before deleting a template, ensure that automatic activation is not used by the application assigned the template in question, unless that outcome is desired.

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

Application connection properties are used to create application connections and application connection templates.

Apple Push Notification Properties

Apple push notification properties allow iOS users to install client software on their devices. This process requires you to create a different e-mail activation message using the appropriate push notification properties.

- **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`.
- **Delivery Threshold –** the frequency, in minutes, with which groupware notifications are sent to the device. Valid values: 0 – 65535. Default: 1.
- **Sounds –** indicates if a sound is a 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

Application Settings

Application settings display details that identify the Application Identifier, Domain, Security Configuration, and Customization Resource of an application connection template

- **Automatic Registration Enabled –** the value is set to **True** when the application connection registration is carried out automatically.
- **Application Identifier –** the application identifier registered on SCC.
- **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.
  - A string of asterisks indicates multiple application connections; the property values are the same, but the customization resource bundle values are different.
  - Blank means no customization resource bundles are assigned.

  Note: Application configuration is only used for OData SDK clients (Android and iOS), and is not used for Hybrid Web Container connections.
- **Domain –** the domain selected for the connection template.

  The domain is not required when automatic registration is enabled.
- **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.

BlackBerry Push Notification Properties

BlackBerry push notification properties allow BlackBerry users to install messaging client software on their devices.

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
Delivery threshold	The minimum amount of time the server waits to perform a push notification to the device since the previous push notification (in minutes). This controls the maximum number of push notifications sent in a given time period. For example, if three push notifications arrive 10 seconds apart, the server does not send three different push notifications to the device. Instead they are sent as a batch with no more than one push notification per X minutes (where X is the delivery threshold). Acceptable values: 0 – 65535. Default: 1
Push listener port	The push listener port reported by the device on which it listens for notifications. This port is automatically assigned by the client. For example, if there is another application already listening on this port, a free port is searched for. Default: 5011
Device PIN	Every Blackberry device has a unique permanent PIN. During initial connection and settings exchange, the device sends this information to the server. Unwired 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
BES Notification Name	The BES server to which this device's notifications are sent. In cases where there are multiple BES servers in an organization, define all BES servers.

Connection Properties

Connection properties define the connection information for a client application so it can locate the appropriate Unwired 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 Unwired 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 Unwired 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 Unwired 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 the *Security* guide.
- **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.

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 workflow-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.

Device Advanced Properties

Advanced properties set specific behavior for messaging devices.

- **Relay Server URL Prefix –** the URL prefix to be used when the device client is connecting through Relay Server. The prefix 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 prefix must match what the administrator configures in the URL suffix. Otherwise the connection fails. Test these value by using the Diagnostic Tool command line utility. 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.

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.

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.
- **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 back-end URL where the application service document is available
- **Push Endpoint –** the server URL where all the notifications are forwarded to.

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 Unwired 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.

Troubleshoot the Sybase Control Center

Troubleshoot issues that arise in Sybase Control Center for Unwired Platform.

Using Sybase Control Center to Troubleshoot Unwired Platform

Problem: Unwired Platform is not functioning properly or exhibits abnormal behaviour.

Consult these Sybase Control Center sources to find useful information to help you troubleshoot Unwired Platform issues:

1. Review the server log – view server errors, warnings, and general information to identify problems. Access the Server node in the left navigation tree of Sybase Control Center to view server log data.
2. Review domain logs – if domain logging is enabled, view domain logs in each Domains > *<DomainName>* > Log node of Sybase Control Center. Aggregated log data in the console makes domain information readily accessible and actionable.
3. Review monitoring data – access the Monitoring node in the left navigation tree of Sybase Control Center to view monitoring data on the following components of Unwired Platform: replication-based synchronization, messaging-based synchronization, messaging queue, data change notifications, device notifications, packages, users, and cache. See *System Diagnostics* in *System Administration* .
4. Review Application Connection status – access the Applications node in the left navigation pane of Sybase Control Center to view application connection information in the right pane.

   Note: You can also view domain-level Application Connection status – navigate to the domain then select **Applications** in the left navigation pane, and view application connection information in the right pane.
5. Review package client logs – access the Client Log tab of the Packages > *<PackageName>* node in Sybase Control Center to view data about client application operations for all devices subscribed to a package. This information allows you to track errors and identify performance issues.
6. Review MBO and operation history – access the History tab for both the MBO and operation nodes of a package in Sybase Control Center to review error history during synchronizations and operation replays.

Collecting Administration Performance Data for Troubleshooting

Problem: You need to collect performance data to troubleshoot performance issues in Sybase Control Center for Unwired Platform administrative options.

Solution: Set up the `<UnwiredPlatform_InstallDir>\SCC-XX\log\executionTime.log`, which provides information on the length of time taken to complete operations in Sybase Control Center. Sybase Product Support and Engineering teams can use this information to diagnose the source of your performance issues. To set up this log file:

1. Open `<UnwiredPlatform_InstallDir>\SCC-XX\plugins\com.sybase.supadminplugin\agent-plugin.xml`.
2. Add the following line to the file under the `<properties>` element:

   ```
   <set-property property="log_MO_method_execution_time"
   value="enable_log_mo_method_execution_time" />
   ```
3. Open `<UnwiredPlatform_InstallDir>\SCC-XX\conf\log4j.properties`.
4. If you are experiencing log truncation issues, edit the following lines to change the default values for maximum file size (default: 5MB) and maximum backup index (default: 10 files) to the values shown in this example:

   ```

   ## file appender (size-based rolling)
   log4j.appender.executionTime=org.apache.log4j.RollingFileAppender
   log4j.appender.executionTime.File=${com.sybase.ua.home}/log/
   executionTime.log
   log4j.appender.executionTime.layout=org.apache.log4j.PatternLayou
   t
   log4j.appender.executionTime.layout.ConversionPattern=%d [%-5p]
   [%t] %c.%M(%L) - %m%n
   log4j.appender.executionTime.MaxFileSize=50MB
   log4j.appender.executionTime.MaxBackupIndex=20
   ## log MO method execution time
   log4j.logger.com.sybase.uep.sysadmin.management.aop=INFO,executio
   nTime
   ```
5. Restart SCC.

   The `executionTime.log` file now appears in the `<UnwiredPlatform_InstallDir>\SCC-XX\log` folder.

Use this log file to diagnose and analyze performance problems. For more information on configuring the `agent-plugin.xml` configuration file, search for *Agent Plugin Properties Reference* in the *System Administration* guide.

You can also use the Adobe Flex log to track performance in Sybase Control Center. To access Flex-side logging, highlight the resource in the Perspective Resources view and select View Log to show the user interface time for each activity. Alternately:

1. Modify the `<UnwiredPlatform_InstallDir>\SCC-XX\plugins\com.sybase.supadminplugin\agent-plugin.xml` file as indicated in step 2, above.
2. Restart SCC.
3. Log in and perform your regular administrative tasks.
4. View the execution time indicators for these operations in the cookie file `supatcookie.sol`. The location of this file varies depending on your operating system:

Operating System	Location
Windows XP	`C:\Documents and Settings\<username>\Application Data\Macromedia\Flash Player\#SharedObjects`
Windows Vista	`C:\Users\<username>\AppData\Roaming\Macromedia\Flash Player\#SharedObjects`
Macintosh OS X	`/Users/<username>/Library/Preferences/Macromedia/Flash Player/#SharedObjects`
Linux	`/home/<username>/.macromedia/Flash_Player/#SharedObjects`

5. Analyze the log using your preferred method of data analysis.

Sybase Control Center Management Tier Issues

Review this list of documented general issues for Sybase Control Center and its server management-related services.

Launching SCC Results in Rounded Rectangle Box or Empty Console Screen

Problem: When you launch Sybase Control Center, a rounded rectangular box appears instead of the administration console, or the console displays a gray or empty screen.

Explanation: The Adobe Flash Player version is older than the minimum version supported by SCC.

Solution: Upgrade your Flash Player version to the latest version. For more information on software prerequisites, see *Supported Hardware and Software*.

Sybase Control Center Windows Service Fails to Start

Problem: When starting the Sybase Control Center *X.X*service, it takes a long time before failing, and the service manager displays a message that the service startup has timed out.

The `<UnwiredPlatform_InstallDir>\SCC_X-X\log\agent.log` shows the following message:

Explanation: This problem usually occurs when the Sybase Control Center repository database log file is out of sync with the repository database. A related symptom is the message `SQL Login Failure` in the Sybase Control Center repository log file.

Solution 1: Review `<UnwiredPlatform_InstallDir>\SCC-X_X\services\Repository\scc_repository.log` log for any issues with the database transaction log file during startup. If the transaction log could not be processed, the database cannot start, and consequently nor can the Sybase Control Center service. Resolve this error by:

1. Creating a backup of `<UnwiredPlatform_InstallDir>\SCC-X_X\services\Repository\scc_repository.log`.
2. Deleting the `<UnwiredPlatform_InstallDir>\SCC-X_X\services\Repository\scc_repository.log` file and restarting the Sybase Control Center service.

Solution 2: Review `<UnwiredPlatform_InstallDir>\SCC-X_X\services\Repository\scc_repository.log` log for any failures in database transaction and/or recovery. Resolve this error by temporarily configuring the repository database (-f) to start without a transaction log:

1. Log out of Sybase Control Center and then shutdown Sybase Control Center service.
2. Open command prompt window, and run the following command:

   ```
   C:\Sybase\SCC-3_2\services\SccSADataserver\sa
   \bin_windows32\dbsrv11.exe -n scc_repository -o C:\Sybase
   \SCC-3_2\services\Repository\scc_repository.slg -f -m -qi -
   qw -sb 0 -gn 100 -gm 500 -zl -zp -x TCPIP{port=3638} C:
   \Sybase\SCC-3_2\services\Repository\scc_repository.db
   ```
3. Delete the `<UnwiredPlatform_InstallDir>\SCC-X_X\services\Repository\scc_repository.log` file using Windows Explorer.
4. Restart the Sybase Control Center service.

Sybase Control Center Windows Service Deleted

Problem: the Sybase Control Center *X.X* windows service was inadvertently deleted, so Sybase Control Center is unavailable.

Solution: Re-create the Windows service with the following command:

```
UnwiredPlatform_InstallDir\SCC-X_X\utility\ntautostart\release
\sccservice.exe -install
```

Sybase Control Center Fails to Start

Problem: The Sybase Control Center server does not start.

This problem occurs when the host name cannot be resolved or the IP address of the machine has changed since the product installation. This troubleshooting topic applies only when either of these scenarios is true.

Solution 1: Change the host name to its IP address in the Sybase Control Center `service-config.xml` file:

1. From the command line, verify the host name by running `nslookup<`*`hostname`* `>`.
2. If the DNS server cannot resolve the host name, edit the colocated `<`*`UnwiredPlatform_InstallDir`*`>\`*`SCC-XX`*`\services\RMI\service-config.xml` file:
   1. Log out of Sybase Control Center.
   2. Stop the Sybase Control Center X.X service.
   3. Open `<`*`UnwiredPlatform_InstallDir`*`>\`*`SCC-XX`*`\services\RMI\service-config.xml`.
   4. Locate this line: `<set-property property="address" value="<`*`hostname`*`>" />`.

      If the line does not exist, add it under the `<properties></properties>` element in the file.
   5. Change the value from the host name to the IP address of the host computer. If the IP address is already used, ensure it is valid (especially if the IP address has recently been changed).
   6. Restart the Sybase Control Center X.X service.
   7. Log in to Sybase Control Center and proceed with your administrative tasks.

Second Sybase Control Center Fails to Start

Problem: Cannot start a second co-existing Sybase Control Center in a deployment environment.

Explanation: When multiple versions of Sybase Control Center co-exist on a single machine, if the older version is already using the default port number, the new version of Sybase Control Center uses another port number, such as 8285. If the configuration files have not been updated, this may cause port conflicts.

Solution: Check the port numbers, and check the configuration files to make sure the configuration is correct. See the topic *Port Number Reference* in the *System Administration* guide. If the configuration is correct, you may need to start the second version of Sybase Control Center manually.

Login Invalid in Sybase Control Center

Problem: Logging in to Sybase Control Center generates an `Invalid Login` message.

Solution:

- Verify Sybase Control Center session validity – ensure that the current Sybase Control Center session is active. If the session is frozen or expired, refresh the page or close the browser and try again.
- Verify server-side configuration by trying to connect to Unwired Server from Unwired WorkSpace (requires creating Unwired Server Connection Profile with proper user name and password among other things).
- Check `<UnwiredPlatform_InstallDir>\SCC_XX\log\agent.log` to see if the authentication failed. If so, check the security configuration. By default, Sybase Control Center shares the same configuration as the one used on the local Unwired Server. Ensure that the correct login and password is provided (one that has administrative privilege). See *Enabling Authentication and RBAC for Administration Logins* in the *Security* guide.
- If all services are running, check the `<UnwiredPlatform_InstallDir>\SCC_XX\log\agent.log` for an error message containing text similar to the following:

  ```
  Failed to authenticate user 'supAdmin' (Failed to connect
  to service:jmx:rmi:///jndi/rmi://eas3w03.sybase.com:9999/
  agent, probably because the agent is protected and requires
  credentials.Security Service Error. Agent service
  exception.)
  ```

  - Ensure that the Sybase Control Center authentication provider configuration is correct, and points to the correct server. See *Enabling Authentication and RBAC for Administration Logins* in the *Security* guide.

Administrator Account is Locked

Problem: An administrator tried logging into Unwired Server from Sybase Control Center multiple times. After receiving multiple instances of the message `Wrong username and password errors`, the message `The account is currently locked. Please contact your server administrator.` is finally displayed.

Explanation: The user has exceeded the threshold for failed login attempts. The platform administrator sets the properties that control the login failure account lock threshold, and the timeout period. When a user exceeds the threshold value, the account is locked for the timeout period.

Solution: A user must wait for the lock timeout value to pass, and then log in again.

For details, see *Creating a Security Configuration* in Sybase Control Center online help.

Browser Refresh (F5) Causes Logout

Problem: Pressing the **F5** key to refresh your browser logs you out of Sybase Control Center.

Solution: Do not use **F5** when you are logged in to Sybase Control Center. Browser refresh does not refresh data inside Sybase Control Center, but refreshes the loaded application or pages in the browser—in this case, the Adobe Flash on which Sybase Control Center is built. Consequently, pressing **F5** logs you out of any servers you are currently logged in to, including Sybase Control Center.

Stale Version of Sybase Control Center After Upgrade

Problem: after upgrading Sybase Unwired Platform and relaunching Sybase Control Center through a Web browser, a stale version of Sybase Control Center loads in the browser.

Explanation: Adobe® Flash® Player caches the earlier version of Sybase Control Center locally, preventing you from logging in to the correct version of Sybase Control Center when accessing the browser.

Solution 1: Clear the Adobe Flash Player cache:

1. In Windows Explorer, navigate to `C:\Documents and Settings\<username>\Application Data\Macromedia\Flash Player\#SharedObjects`, and delete all files in this folder.

   As an alternative to manually deleting files, you can also access the Adobe Flash Player Cache Cleanup URL: *http://www.macromedia.com/support/documentation/en/flashplayer/help/settings_manager07.html*

Solution 2: Only perform this solution if Solution 1 does not solve the problem. Clear browser history:

1. In Microsoft Internet Explorer, select **Tools** > **Internet Options** > **General** > **Delete...** and delete all temporary files, history, cookies, saved passwords, and Web form information.

Sybase Control Center Reports Certificate Problem

When attempting to bring up Sybase Control Center by clicking the SCC link after installation , this message appears: `There is a problem with this website's security certificate`.

Explanation: This can occur when the browser session starts on the same computer as Sybase Control Center. 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 b the client browser.

Solution: Follow browser-specific instructions to accept the certificate into the Windows certificate store. Once the certificate is accepted, you may also need to change the SCC Web

URL to include the network domain name *<yourco.com>* in addition to the host name. That host name in the Web URL must match with the "Issued To" property of the certificate.

Previous Administrator Credentials Used

Problem: You cannot use new credentials to authenticate against a resource in Sybase Control Center. When an administrator enters credentials with the **Remember these credentials for future sessions** option, Sybase Control Center uses those credentials until they are cleared.

Solution: Clear credentials so that Sybase Control Center does not use them for future sessions:

1. Open the Perspective Resources window.
2. Select the resource you want to log in to.
3. From the menu bar, select **Resource > Clear Authentication Parameters** and click **OK**.

You can now authenticate against the resource using new administrator credentials.

Security Error Triggered When Connecting to SCC from Remote Browser

Problem: Connecting to Sybase Control Center from a browser that is remote triggers a security exception.

Solution: Ensure you have a security certificate installed in the Windows security store. See *Setting Up Browser Certificates for Sybase Control Center Connections* in Sybase Control Center online help.

Administrator Login Passes When Provider Is Not Available

Problem: The configured authentication provider is unavailable but administration credentials are still accepted.

Explanation: The administrator login credentials may be cached by Unwired Server.

Solution: If this behavior is undesired, reduce the cache timeout value used by the Unwired Server security domain instance. For details, search for *Authentication Cache Timeouts* in the *Security* guide.

Host Name of Registered Resource Changed But Is Not Updated

Problem: An administrator changes the host name property of a registered resource; but in Sybase Control Center, the old host name is still used and the management console for Unwired Platform does not appear.

Description: If you modify the resource properties for an Unwired Server in Sybase Control Center, the new host name or IP address is not used in establishing a connection to the server.

Solution: After changing the host name property of the resource, in the Perspective Resources view, right-click the resource and select **Authenticate** to update resource connection properties. You can then launch the management console successfully.

Poor Sybase Control Center Performance after Upgrade

Problem: After upgrading to the latest version of Sybase Unwired Platform, Sybase Control Center performance is poor.

Explanation: This may indicate that Flash Player cache from the previous version of Sybase Control Center is filled and slowing down performance.

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

Sybase Control Center Communication with Unwired Server Fails

Problem: While using Sybase Control Center, a `Communication with Unwired Server failed` appears.

Explanation: Sybase Control Center cannot connect to the Unwired Server and displays this error message. To confirm this issue, open the Sybase Control Center `<SCC_HOME>\log\gateway.log` and look for `org.omg.CORBA.COMM_FAILURE`, `com.sybase.djc.rmi.iiop.BadMagicException`, or `org.omg.CORBA.MARSHAL` entries.

Solutions:

1. Ensure the protocol and port used by both the Unwired Server management port and the Sybase Control Center managed resource registration entry match. For information about validating and changing these properties, see *Adding or Updating Unwired Server Registration Properties* in the Sybase Control Center online help.
2. Validate that the configured port is the Unwired Server management port of 2000 or 2001. Sybase recommends that you not change these default values. If you have changed them and the connection fails, update the managed resource connection property to use the default.

   For more information about ports, see Port Number Reference in *System Administration*. For more information about validating and changing this port, see *Registering a Resource as an SCC Managed Resource* in the Sybase Control Center online help.
3. If the management security profile now uses SSL mutual authentication (`mutual_auth`), validate that you have installed certificates for mutual authentication into both Sybase Control Center's and Unwired Server's keystore. If each component

doesn't have the opposite set of certificates, mutual authentication fails. Either install the missing certificates if mutual authentication is required, or use the following procedure to recover from this scenario:

1. If Unwired Server also has a standard management (non-secure) port available, you can connect to that port by updating the Sybase Control Center resource (localhost) port number property and setting secure to "No". See *Adding or Updating Unwired Server Registration Properties* in Sybase Control Center online help.
2. If Unwired Server doesn't have the standard management port enabled, then update the **securityProfile** property value in the `<UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\UnwiredServer\Repository\Instance\com\sybase\djc\server\SocketListener\{ServerName}_iiops1.properties`file to use the "default" profile, and restart Unwired Server.

For information about installing certificates, see *Changing Installed Certificates Used for Unwired Server and Sybase Control Center HTTPS Listeners*, in the *Security* guide. For information about changing the authentication method used by the security profile, see *Creating an SSL Security Profile in Sybase Control Center* in Sybase Control Center online help.

Server Tier Administration Issues

Review this list of documented issues for Unwired Server or its internal synchronization services configured and administered by Sybase Control Center.

Server List Not Retrieved

Problem: No list of Unwired Servers displays in Sybase Control Center. Instead, an `Error Retrieving Server List` message appears in the left navigation pane.

Scenario 1: No other error message appears.

If this is the case, one of the following explanations may apply:

- You are attempting to connect to a remote server that is not properly registered in Sybase Control Center.

  Solution: Manually register the remote server. By default, only Unwired Servers installed to the same host computer are automatically registered with Sybase Control Center. See *Getting Started with Unwired Server Administration* in the Sybase Control Center online help. If you have recently made changes to the environment, for example, by modifying server resource properties (login, password, host name, IP address, or port number), ensure that you reauthenticate after making the changes.
- Jetty caching in Sybase Control Center prevents the console from displaying the server tree. This is indicated by 404 errors in both the console URL and `<UnwiredPlatform_InstallDir>\SCC-X_X\services\EmbeddedWebContainer\log\http-service.log` (the HTTP access log).

Solution:

1. Close Sybase Control Center.
2. Stop Sybase Control Center X.X Service.
3. Delete the contents of: `<`*`UnwiredPlatform_InstallDir`*`>\SCC-`*`X_X`*`\services\EmbeddedWebContainer\container\Jetty-`*`X.X.XX`*`\work`.
4. Restart Sybase Control Center *X.X* service.

Scenario 2: The right administration pane shows an `Authentication has failed error` message.

If this is the case, one of the following explanations may apply:

- You have not performed the "Authenticate" step in Sybase Control Center after registering the resource or changing their credentials.

  Solution: In the Perspective Resources view, right click the server name and select **Authenticate**. In the default configuration, if you have used "supAdmin" to log in to Sybase Control Center, select **Use my current SCC login**.
- The server IP may have changed.

  Solution: Update server resource properties, and repeat the "Authenticate" step described above. See the topic *Sybase Control Center Fails to Start*.

Scenario 3: The right administration pane shows a `Connection unknown. Ensure Server is running....` message.

If this is the case, one of the following explanations may apply:

- Unwired Server responded with an exception indicating a problem on the server.

  Solution: Check `<`*`UnwiredPlatform_InstallDir`*`>\UnwiredPlatform\Servers\UnwiredServer\logs\<`*`hostname`*`>-server.log` for details.
- The Sybase Control Center security provider is down or a system condition prevents Sybase Control Center from authenticating the user for administration access.

  Solution: Ensure that the security provider is running and that its host is reachable from the Sybase Control Center host.

Scenario 4: In some rare cases, the connection between Sybase Control Center and Unwired Server cannot be established after trying the previous recommendations.

Solution: You may need to stop and restart the Sybase Control Center *X.X* windows service. After stopping the window service, make sure the process `uaservices.exe` is not running (or stop it from Windows task manager). Then log in to Sybase Control Center again.

Scenario 5: This may happen if you upgraded Sybase Unwired Platform to a newer version, and changed the server host name.

Solution: You need to complete some extra steps:

1. Change the listener prefix of httpListeners and iiopListeners for the new hostname in the new server's properties file:

   ```
   Repository\Instance\com\sybase\djc\server\ApplicationServer
   \default.properties,  <new_hostname>.properties
   ```
2. In `Repository\Instance\com\sybase\djc\server\SocketListner\*.properties`, rename all the `<old_hostname>_<protocol>.properties` into `<new_hostname>_<protocol>.properties`.
3. Use dbisqlc to update the table: `cluster_installation in clusterdb, update cluster_installation set hostname='<new_hostname>' where hostname='<old_hostname>'`.

Unwired Server Fails to Start

Problem: Starting Unwired Server from Windows services or the desktop shortcut fails.

Solution:

1. Ensure that the server license is valid and has not expired.
2. Open Windows services to check that the services Unwired Server depends on for start-up are running properly. Identify dependencies by right-clicking the service and selecting **Properties.**
3. Check `<UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\UnwiredServer\log\<serverName>-server.log` for error messages indicating the nature of Unwired Server start-up issues.
4. Check `<UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\UnwiredServer\log\bootstrap**.log` for possible license errors.

Error in Listing Application Connections and ADMIN_WEBSERVICE_INVOCATION_ERROR in gateway.log

Problem: This message may indicate that an Unwired Server administrative component is not running.

If users report a problem listing application connections in Sybase Control Center, check for this error message in the Sybase Control Center `gateway.log` file:

```
com.sybase.uep.sysadmin.management.mbean.UEPAdminException:
com.sybase.uep.admin.client.AdminException:
ADMIN_WEBSERVICE_INVOCATION_ERROR:java.security.PrivilegedActionExc
eption: com.sun.xml.internal.messaging.saaj.SOAPExceptionImpl:
Message send failed
javax.management.MBeanException:
```

Explanation: Usually this occurs when there is a conflict on the currently configured port for the administration web service or a component of Sybase Unwired Server service went down for some reason.

One way to verify availability of the Web service is by accessing the following URL from the host where Sybase Unwired Platform is installed: *http://localhost:5100/MobileOffice/Admin.asmx*.

Note: This link works from the host where Sybase Unwired Platform is installed, using the correct Messaging port. The default Messaging port is 5100, but this may vary depending on your configuration.

Solution 1: Check Windows Application Event log for any error reported there. If the service is configured to run with a domain account and the password has been changed, you will need to update the password.

Solution 2: Make sure the administration Web service is up and running, and correctly configured. Review *Cannot Access Applications Tab and Web Service Error* in *Troubleshooting* to reconfigure the port in case of conflict with existing port.

Starting or Restarting a Remote Server from Sybase Control Center Fails

Problem: After you have registered a remote server in Sybase Control Center, you cannot start or restart the server.

If the DNS server cannot resolve the host name of the machine on which the remote Unwired Server is installed, or if the host has no internal DNS server, you cannot start, stop, or restart that Unwired Server using your local instance of Sybase Control Center. Because this network communication relies on name resolution, you must ensure that DNS is set up properly to successfully control a remote Unwired Server.

Before attempting the following solutions, verify that:

1. Sybase Control Center is running on the remote host.
2. A network connection can be established between your Sybase Control Center host and the Sybase Control Center agent on the remote server's host.

If the DNS server cannot establish a connection, try the following:

Solution 1: Repair the network DNS server setup. If you or your network administrator cannot modify the DNS, use solution 2.

Solution 2: Change the host name to its IP address in the Sybase Control Center `service-config.xml` file:

- If you cannot resolve the local host name, modify the file on the local instance of Sybase Control Center.
- If you cannot resolve the remote host name, modify the file on the remote instance of Sybase Control Center.
- If you cannot resolve both the remote and local host names, modify both files.

1. From the command line, verify the host name by running `nslookup<`*`hostname`* `>`.

2. If the DNS server cannot resolve the host name, edit the colocated `<`*`UnwiredPlatform_InstallDir`*`>\`*`SCC-XX`*`\services\RMI\service-config.xml` file:
    a. Log out of Sybase Control Center.
    b. Stop the Sybase Control Center X.X service.
    c. Open `<`*`UnwiredPlatform_InstallDir`*`>\`*`SCC-XX`*`\services\RMI\service-config.xml`.
    d. Locate this line: `<set-property property="address" value="<`*`hostname`*`>" />`.

       If the line does not exist, add it under the `<properties></properties>` element in the file.
    e. Change the value from the host name to the IP address of the host computer. If the IP address is already used, ensure it is valid (especially if the IP address has recently been changed).
    f. Restart the Sybase Control Center X.X service.
    g. Log in to Sybase Control Center and proceed with your administrative tasks.

If the DNS server resolves the host name, but the problem persists, check that both:

- The remote host on which Unwired Platform and Sybase Control Center are installed can receive UDP multicasts from the local host on which Sybase Control Center is installed, and
- The remote instance of Sybase Control Center uses RMI port 9999.

Solution 3: Make sure the `hosts` file includes complete entries for each node in the Unwired Server cluster.

1. On each Unwired Server host, edit the `hosts` file, located at: `C:\WINDOWS\system32\drivers\etc`
2. Add entries to identify the IP address and fully qualified network name of every other node in the Unwired Server cluster.

Port Conflict Issues

Problem: You have identified a Sybase Control Center *X.X* service port conflict.

Solution:

1. Identify the service with the port conflict in `<UnwiredPlatform_InstallDir>\SCC-`*`X_X`*`\log\agent.log`.
2. Use a text editor to open `<UnwiredPlatform_InstallDir>\SCC-`*`X_X`*`\Services\<`*`Servicename`*`>\service-config.xml`.
3. Change the port to an available port number.
4. Save and close the file.

Search for *Port Number Reference* in *System Administration* for more information.

Unexpected Listener Startup or Connection Errors

Problem: You encounter unexpected listener startup or connection errors for Unwired Platform components. This is usually seen when Sybase Unwired Server is installed on a host in DMZ (De-Militarized Zone) within the internal and external firewalls.

Solution:

1. Verify that the TCP/IP filtering restriction is not in effect on the host machine.

   To do so on Windows XP, navigate to: **Control Panel > Network Connections > Local Area Connection 1 > Properties > General tab > Internet Protocol (TCP/IP) > Properties > General tab > Advanced > Options tab > TCP/IP filtering > Properties**
2. In TCP/IP Filtering, check to make sure the Enable TCP/IP Filtering (All Adopters) checkbox is not selected. This enables all Sybase Unwired Platform infrastructure ports.

   If you do choose to select it, be sure to select Permit All for TCP Ports to enable all Sybase Unwired Platform infrastructure ports. These ports are documented in the Installation Guide.
3. Click **OK** to close each window and save your changes.
4. You can change “Local Area Connection 1” to the network connection name being used on the machine.
5. Make sure users are not using third party port blockers, like McAfee Antivirus.

Refreshing Server Configuration Displays Only Partial Updates

Problem: The Refresh button in the Server Configuration node does not display correct properties or values, despite changes being made and saved. Updates consequently appear to have been lost. In some scenarios, when you save the Server Configuration, it fails with the message `Save Failed`.

Scenario 1: After restarting Unwired Server, refreshing the server configuration displays the first saved change, but not subsequent saved updates. The message `Save Failed` appears in the administration console after you attempt to save an update.

In this scenario, the second save was likely unsuccessful. The message `Save Failed` indicates a conflict with the first set of updates.

Cumulative saved changes are applied successfully upon server restart only if these updates do not conflict. Attempting to save two conflicting sets of changes fails.

Solution: Inject a server restart in between each saved change to ensure that the required updates are propagated across the server.

Scenario 2: After restarting Unwired Server, refreshing the server configuration displays the final saved update, but not previous ones.

The refresh action following saved configuration changes must be used in conjunction with an Unwired Server restart. Refreshing the server configuration displays the latest successfully saved configuration information.

If you click Refresh in between two sets of saved changes, only the most recent saved updates are applied during a server restart, as in the following workflow:

1. Make the first change.
2. Save the configuration.
3. Refresh the configuration.
4. Make the second change.
5. Save the configuration.
6. Restart the server.
7. Refresh the configuration.

In this sequence, only the second set of changes in step 4 are committed and consequently displayed as the current set of properties used by Unwired Server.

Solution: If you refresh the configuration after saving updates to it, restart Unwired Server immediately to apply those changes before making another set of updates. Otherwise, the first set of configuration changes will be lost. The Refresh button allows you to then validate that those changes are applied and used by Unwired Server. For details on how to refresh the server in the correct sequence, see *Saving and Refreshing an Unwired Server Configuration* in the Sybase Control Center online help.

Users Connect with Old Credentials

Problem: A user changes password in the backend security system, but can still authenticate with the previous password when connecting to Unwired Server.

Description: Unwired Server securely caches authenticated login credentials (1 hour by default), so that subsequent connection requests using the same credentials are not sent to the underlying security provider until the login cache timeout is reached. However, if the same user uses changed credentials, the authentication request us sent to the underlying security provider. The authorization outcome is not cached and always delegated to the security provider in the security configuration.

Solution: To reduce the cache period, decrease the default authentication cache timeout for a security configuration using Sybase Control Center (go to the Cluster > Security > <security configurationname> > Settings tab). Setting the property to 0 results in disabling the authentication caching (not recommended for performance reasons).

AuthorizationException Displays Instead of Status

The SCC administration console left-pane tree structure is not complete, and an AuthorizationException is reported..

Explanation: This may happen if the SCC administration console internal network communications are not working properly.

Solution:

1. Close the Internet Explorer session.
2. Relaunch the SCC administrative console.
3. Log in as usual.

   The internal network connection is resumed by restarting, so the tree displays information and status properly.

Saving Server Configuration Fails Due to Certificate Validation Error

Problem: Saving the server configuration after property updates yields this error: "[com.sybase.sup.admin.server.configuration.RuntimeServerConfigurationHandler] Invalid configuration object for: SyncServerConfiguration. Message : 'certificate validation failed. Update did not happen.'"

Solution: The message suggests that the server certificate has expired. Update the certificate file to a non-expired version, and try to save again.

Unknown Server Error Message

Problem: An internal server error occurs.

Solution: Check the logs for more details. Start by looking at the SCC log file `\SCC-`*`X_X`*`\log\gateway.log` for the error message that occurred when user interface displayed the message. In most cases, the error is an unexpected failure condition in the server, and further details can be obtained by reviewing the `\UnwiredPlatform\Servers\UnwiredServer\logs\{ServerName}-server.log`. If that does not help, contact your Sybase Unwired Platform technical support representative.

Application and Application User Management Issues

Review this list of documented issues for applications or application users managed by Sybase Control Center.

Wrong Application for Code Error

Problem: Application registration using a Windows Mobile emulator appears successful in Sybase Control Center, but the application log shows a `Wrong Application for Code` error when the application attempts to connect to Unwired Server.

This error occurs when you:

- Hard reset a Windows Mobile device emulator,
- Close an emulator without saving the emulator state, or
- Uninstall and reinstall the Unwired Server client software on the device.

Explanation: Because emulators do not generate unique application IDs, the Unwired Server messaging software on the device creates an application ID during installation and stores it in the emulator application registry. After registration, this permanent link between the emulator and the application ID must remain.

Hard resetting the emulator, closing the emulator without saving the emulator state, or uninstalling and reinstalling the Unwired Server client software purges the device registry and breaks the link between Unwired Server and the device software. When you attempt to reconnect, Unwired Server creates a new application ID for the device. Without the original application ID, the server cannot identify the device emulator, and therefore, cannot establish a relationship between the application and the activation code.

To avoid this problem so that the emulator and server remain synchronized, always save the emulator state before you close the emulator, and refrain from hard resetting the emulator, or uninstalling and reinstalling the client software.

Note: Before saving the state of an emulator, always uncradle the emulator using the Device Emulation Manager. This allows the device emulator to be cradled when the save image is loaded and used in the future.

Solution: Reconnect the emulator by either:

1. Deleting the original application from Unwired Server, then reregister the application, or
2. Reregistering the application

User Name of Registered Application Connection Not Displayed

Problem: The configured user name of a registered application connection is not displayed when you later review the properties for a device in Sybase Control Center. The **Application Connections** tab shows other properties but not the user name.

Explanation: The user name used for a application connection registration is not stored or handled as an application property.

Solution: To view the user name of the registered application in Sybase Control Center:

1. In the left navigation pane, click the **Applications** node.
2. In the right administration pane, click the **Application Users** tab.
3. In the table of registered users, for the user.
4. You can also select the **Application Connections** tab, and check the users properties.

Internal Server Error When Clicking Applications

Problem : Once logged into Sybase Control Center, the administrator clicks Applications in the navigation pane, and an `Internal server error` message is displayed.

After receiving this error, the administrator is further unable to register any applications because the **OK** button remains disabled.

Solution:

1. Validate the error:
   1. Open `<`*`UnwiredPlatform_InstallDir`*`>\SCC-`*`X_X`*`\log\gateway.log`.
   2. Look for this error: `Caused by: com.sybase.uep.sysadmin.management.exception.ImoWsException: An error occurred loading a configuration file: Attempted to read or write protected memory. This is often an indication that other memory is corrupt.`
2. Validate that the Sybase Unwired Server service is running and there are no errors being reported in the Windows Application event log by that service.
3. Validate that the Messaging Server Administration Web Service is running:
   1. Open a Web browser.
   2. Open http://localhost:5100/MobileOffice/admin.asmx.
   3. Select the **GetDeviceList2** method, then click **Invoke**.
   4. Check whether a valid XML response returns.
4. If anything in steps 1-3 is unexpected, you may have an installation or configuration issue. Confirm this by:
   1. Restarting the Sybase Unwired Server service.
   2. Once available, repeat steps 2-3.
      - Otherwise, open Sybase Control Center, and click Applications to try registering an application again.
5. If you still get the same error and same behavior, contact Sybase Support.

Glossary

Defines terms used in Sybase Control Center documentation.

Glossary: Sybase Unwired Platform

Defines terms for all Sybase Unwired Platform components.

administration perspective – Or administration console. The Unwired Platform administrative perspective is the Flash-based Web application for managing Unwired Server. *See* Sybase Control Center.

administrators – Unwired Platform users to which an administration role has been assigned. A user with the "SUP Administrator" role is called a "platform administrator" and a user with the "SUP Domain Administrator" role is called a "domain administrator". These administration roles must also be assigned SCC administration roles to avoid having to authenticate to Sybase Control Center in addition to Unwired Server:

- A domain administrator only requires the "sccUserRole" role.
- A platform administrator requires both the "sccAdminRole" and "sccUserRole" roles.

Adobe Flash Player – Adobe Flash Player is required to run Sybase Control Center. Because of this player, you are required to run Sybase Control Center in a 32-bit browser. Adobe does not support 64-bit browsers.

Advantage Database Server® – A relational database management system that provides the messaging database for Sybase Unwired Platform. *See* messaging database.

Afaria® – An enterprise-grade, highly scalable device management solution with advanced capabilities to ensure that mobile data and devices are up-to-date, reliable, and secure. Afaria is a separately licensed product that can extend the Unwired Platform in a mobile enterprise. Afaria includes a server (Afaria Server), a database (Afaria Database), an administration tool (Afaria Administrator), and other runtime components, depending on the license you purchase.

application – In Unwired Server (and visible in Sybase Control Center), and application is the runtime entity that can be directly correlated to a native or mobile workflow application. The application definition on the server establishes the relationship among packages used in the application, domain that the application is deployed to, user activation method for the application, and other application specific settings.

APNS – Apple Push Notification Service.

application connection – A unique connection to the application on a device.

application connection template – a template for application connections that includes application settings, security configuration, domain details, and so forth.

application node – In Sybase Control Center, this is a registered application with a unique ID. This is the main entity that defines the behavior of device and backend interactions.

application registration – The process of registering an application with Sybase Unwired Platform. Registration requires a unique identity that defines the properties for the device and backend interaction with Unwired Server.

artifacts – Artifacts can be client-side or automatically generated files; for example: `.xml`, `.cs`, `.java`, `.cab` files.

availability – Indicates that a resource is accessible and responsive.

BAPI – Business Application Programming Interface. A BAPI is a set of interfaces to object-oriented programming methods that enable a programmer to integrate third-party software into the proprietary R/3 product from SAP®. For specific business tasks such as uploading transactional data, BAPIs are implemented and stored in the R/3 system as remote function call (RFC) modules.

BLOB – Binary Large Object. A BLOB is a collection of binary data stored as a single entity in a database management system. A BLOB may be text, images, audio, or video.

cache – The virtual tables in the Unwired Server cache database that store synchronization data.*See* cache database.

cache group – Defined in Unwired WorkSpace, MBOs are grouped and the same cache refresh policy is applied to their virtual tables (cache) in the cache database

cache partitions – Partitioning the cache divides it into segments that can be refreshed individually, which gives better system performance than refreshing the entire cache. Define cache partitions in Unwired WorkSpace by defining a partition key, which is a load argument used by the operation to load data into the cache from the enterprise information system (EIS).

cache database – Cache database. The Unwired Server cache database stores runtime metadata (for Unwired Platform components) and cache data (for MBOs). *See also* data tier.

CLI – Command line interface. CLI is the standard term for a command line tool or utility.

client application – *See* mobile application.

client object API – The client object API is described in the *Developer Guide: BlackBerry Native Applications*, *Developer Guide: iOS Native Applications*, and *Developer Guide: Windows and Windows Mobile Native Applications*.

cluster – Also known as a server farm. Typically clusters are setup as either runtime server clusters or database clusters (also known as a data tier). Clustering is a method of setting up redundant Unwired Platform components on your network in order to design a highly scalable and available system architecture.

cluster database – A data tier component that holds information pertaining to all Unwired Platform server nodes. Other databases in the Unwired Platform data tier includes the cache, messaging, and monitoring databases.

connection – Includes the configuration details and credentials required to connect to a database, Web service, or other EIS.

connection pool – A connection pool is a cache of Enterprise Information System (EIS) connections maintained by Unwired Server, so that the connections can be reused when Unwired Server receives future requests for data.

For proxy connections, a connection pool is a collection of proxy connections pooled for their respective back-ends, such as SAP Gateway.

connection profile – In Unwired WorkSpace, a connection profile includes the configuration details and credentials required to connect to an EIS.

context variable – In Unwired WorkSpace, these variables are automatically created when a developer adds reference(s) to an MBO in a mobile application. One table context variable is created for each MBO attribute. These variables allow mobile application developers to specify form fields or operation parameters to use the dynamic value of a selected record of an MBO during runtime.

data change notification (DCN) – Data change notification (DCN) allows an Enterprise Information System (EIS) to synchronize its data with the cache database through a push event.

data refresh – A data refresh synchronizes data between the cache database and a back-end EIS so that data in the cache is updated. *See also* scheduled data refresh.

data source – In Unwired WorkSpace, a data source is the persistent-storage location for the data that a mobile business object can access.

data tier – The data tier includes Unwired Server data such as cache, cluster information, and monitoring. The data tier includes the cache database (CDB), cluster, monitoring, and messaging databases.

data vault – A secure store across the platform that is provided by an SUP client.

deploy – (Unwired Server) Uploading a deployment archive or deployment unit to an Unwired Server instance. Unwired Server can then make these units accessible to users via a client application that is installed on a mobile device.

There is a one-to-one mapping between an Unwired WorkSpace project and a server package. Therefore, all MBOs that you deploy from one project to the same server are deployed to the same server package.

deployment archive – In Unwired WorkSpace, a deployment archive is created when a developer creates a package profile and executes the **build** operation. Building creates an archive that 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 Unwired Server.

deployment descriptor – A deployment descriptor is an XML file that describes how a deployment unit should be deployed to Unwired Server. A deployment descriptor contains role-mapping and domain-connection information. You can deliver a deployment descriptor and a deployment unit—jointly called a deployment archive—to an administrator for deployment to a production version of Unwired Server.

deployment mode – You can set the mode in which a mobile application project or mobile deployment package is deployed to the target Unwired Server.

deployment profile – A deployment profile is a named instance of predefined server connections and role mappings that allows developers to automate deployment of multiple packages from Sybase Unwired WorkSpace to Unwired Server. Role mappings and connection mappings are transferred from the deployment profile to the deployment unit and the deployment descriptor.

deployment unit – The Unwired WorkSpace build process generates a deployment unit. It enables a mobile application to be effectively installed and used in either a preproduction or production environment. Once generated, a deployment unit allows anyone to deploy all required objects, logical roles, personalization keys, and server connection information together, without requiring access to the whole development project. You can deliver a deployment unit and a deployment descriptor—jointly called a deployment archive—to an administrator for deployment to a production version of Unwired Server.

development package – A collection of MBOs that you create in Unwired WorkSpace. You can deploy the contents of a development package on an instance of Unwired Server.

device application – *See also* mobile application. A device application is a software application that runs on a mobile device.

device notification – Replication synchronization clients receive device notifications when a data change is detected for any of the MBOs in the synchronization group to which they are subscribed. Both the change detection interval of the synchronization group and the notification threshold of the subscription determine how often replication clients receive device notifications. Administrators can use subscription templates to specify the notification threshold for a particular synchronization group.

device user – The user identity tied to a device.

DML – Data manipulation language. DML is a group of computer languages used to retrieve, insert, delete, and update data in a database.

DMZ – Demilitarized zone; also known as a perimeter network. The DMZ adds a layer of security to the local area network (LAN), where computers run behind a firewall. Hosts running in the DMZ cannot send requests directly to hosts running in the LAN.

domain administrator – A user to which the platform administrator assigns domain administration privileges for one or more domain partitions. The domain administrator has a restricted view in Sybase Control Center, and only features and domains they can manage are visible.

domains – Domains provide a logical partitioning of a hosting organization's environment, so that the organization achieves increased flexibility and granularity of control in multitenant environments. By default, the Unwired Platform installer creates a single domain named "default". However the platform administrator can also add more domains as required.

EIS – Enterprise Information System. EIS is a back-end system, such as a database.

Enterprise Explorer – In Unwired WorkSpace, Enterprise Explorer allows you to define data source and view their metadata (schema objects in case of database, BAPIs for SAP, and so on).

export – The Unwired Platform administrator can export the mobile objects, then import them to another server on the network. That server should meet the requirement needed by the exported MBO.

hostability – *See* multitenancy.

IDE – Integrated Development Environment.

JDE – BlackBerry Java Development Environment.

key performance indicator (KPI) – Used by Unwired Platform monitoring. KPIs are monitoring metrics that are made up for an object, using counters, activities, and time which jointly for the parameters that show the health of the system. KPIs can use current data or historical data.

keystore – The location in which encryption keys, digital certificates, and other credentials in either encrypted or unencrypted keystore file types are stored for Unwired Server runtime components. *See also* truststore.

LDAP – Lightweight Directory Access Protocol.

local business object – Defined in Unwired WorkSpace, local business objects are not bound to EIS data sources, so cannot be synchronized. Instead, they are objects that are used as local data store on device.

logical role – Logical roles are defined in mobile business objects, and mapped to physical roles when the deployment unit that contain the mobile business objects are deployed to Unwired Server.

matching rules – A rule that triggers a mobile workflow application. Matching rules are used by the mobile workflow email listener to identify e-mails that match the rules specified by the administrator. When emails match the rule, Unwired Server sends the e-mail as a mobile workflow to the device that matches the rule. A matching rule is configured by the administrator in Sybase Control Center.

MBO – Mobile business object. The fundamental unit of data exchange in Sybase Unwired Platform. An MBO roughly corresponds to a data set from a back-end data source. The data can come from a database query, a Web service operation, or SAP. An MBO contains both concrete implementation-level details and abstract interface-level details. At the implementation-level, an MBO contains read-only result fields that contain metadata about the data in the implementation, and parameters that are passed to the back-end data source. At the interface-level, an MBO contains attributes that map to result fields, which correspond to client properties. An MBO may have operations, which can also contain parameters that map to arguments, and which determines how the client passes information to the enterprise information system (EIS).

You can define relationships between MBOs, and link attributes and parameters in one MBO to attributes and parameters in another MBO.

MBO attribute – An MBO attribute is a field that can hold data. You can map an MBO attribute to a result field in a back-end data source; for example, a result field in a database table.

MBO binding – An MBO binding links MBO attributes and operations to a physical data source through a connection profile.

MBO operation – An MBO operation can be invoked from a client application to perform a task; for example, create, delete, or update data in the EIS.

MBO relationship – MBO relationships are analogous to links created by foreign keys in a relational database. For example, the account MBO has a field called *owner_ID* that maps to the *ID* field in the owner MBO.

Define MBO relationships to facilitate:

- Data synchronization
- EIS data-refresh policy

messaging based synchronization – A synchronization method where data is delivered asynchronously using a secure, reliable messaging protocol. This method provides fine-grained synchronization (synchronization is provided at the data level—each process communicates only with the process it depends on), and it is therefore assumed that the device is always connected and available. *See also* synchronization.

messaging database – The messaging database allows in-flight messages to be stored until they can be delivered. This database is used in a messaging based synchronization environment. The messaging database is part of the Unwired Platform data tier, along with the cache, cluster, and monitoring databases.

mobile application – A Sybase Unwired Platform mobile application is an end-to-end application, which includes the MBO definition (back-end data connection, attributes, operations, and relationships), the generated server-side code, and the client-side application code.

Mobile Application Diagram – The Mobile Application Diagram is the graphical interface to create and edit MBOs. By dragging and dropping a data source onto the Mobile Application Diagram, you can create a mobile business object and generate its attribute mappings automatically.

Mobile Application Project – A collection of MBOs and client-side, design-time artifacts that make up a mobile application.

mobile workflow packages – Mobile workflow packages use the messaging synchronization model. The mobile workflow packages are deployed to Unwired Server, and can be deployed to mobile devices, via the Unwired Platform administrative perspective in Sybase Control Center.

monitoring – Monitoring is an Unwired Platform feature available in Sybase Control Center that allows administrators to identify key areas of weakness or periods of high activity in the particular area they are monitoring. It can be used for system diagnostic or for troubleshooting. Monitored operations include replication synchronization, messaging synchronization, messaging queue, data change notification, device notification, package, user, and cache activity.

monitoring database – A database that exclusively stores data related to replication and messaging synchronization, queues status, users, data change notifications, and device notifications activities. By default, the monitoring database runs in the same data tier as the cache database, messaging database and cluster database.

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.

multitenancy – The ability to host multiple tenants in one Unwired Cluster. Also known as hostability. *See also* domains.

node – A host or server computer upon which one or more runtime components have been installed.

object query – Defined in Unwired WorkSpace for an MBO and used to filter data that is downloaded to the device.

onboarding – The enterprise-level activation of an authentic device, a user, and an application entity as a combination, in Unwired Server.

operation – *See* MBO operation.

package – A package is a named container for one or more MBOs. On Unwired Server a package contains MBOs that have been deployed to this instance of the server.

palette – In Unwired WorkSpace, the palette is the graphical interface view from which you can add MBOs, local business objects, structures, relationships, attributes, and operations to the Mobile Application Diagram.

parameter – A parameter is a value that is passed to an operation/method. The operation uses the value to determine the output. When you create an MBO, you can map MBO parameters to data-source arguments. For example, if a data source looks up population based on a state abbreviation, the MBO gets the state from the user, then passes it (as a parameter/argument) to the data source to retrieve the information. Parameters can be:

- Synchronization parameters – synchronize a device application based on the value of the parameter.
- Load arguments – perform a data refresh based on the value of the argument.
- Operation parameters – MBO operations contain parameters that map to data source arguments. Operation parameters determine how the client passes information to the enterprise information system (EIS).

personalization key – A personalization key allows a mobile device user to specify attribute values that are used as parameters for selecting data from a data source. Personalization keys are also used as operation parameters. Personalization keys are set at the package level. There are three type of personalization keys: Transient, client, server.

They are most useful when they are used in multiple places within a mobile application, or in multiple mobile applications on the same server. Personalization keys may include attributes such as name, address, zip code, currency, location, customer list, and so forth.

perspective – A named tab in Sybase Control Center that contains a collection of managed resources (such as servers) and a set of views associated with those resources. The views in a perspective are chosen by users of the perspective. You can create as many perspectives as you need and customize them to monitor and manage your resources.

Perspectives allow you to group resources ways that make sense in your environment—by location, department, or project, for example.

physical role – A security provider group or role that is used to control access to Unwired Server resources.

Problems view – In Eclipse, the Problems view displays errors or warnings for the Mobile Application Project.

provisioning – The process of setting up a mobile device with required runtimes and device applications. Depending on the synchronization model used and depending on whether or not the device is also an Afaria client, the files and data required to provision the device varies.

pull synchronization – Pull synchronization is initiated by a remote client to synchronize the local database with the cache database. On Windows Mobile, pull synchronization is supported only in replication applications.

push synchronization – Push is the server-initiated process of downloading data from Unwired Server to a remote client, at defined intervals, or based upon the occurrence of an event.

queue – In-flight messages for a messaging application are saved in a queue. A queue is a list of pending activities. The server then sends messages to specific destinations in the order that they appear in the queue. The depth of the queue indicates how many messages are waiting to be delivered.

relationship – *See* MBO relationship.

relay server – *See also* Sybase Hosted Relay Service.

resource – A unique Sybase product component (such as a server) or a subcomponent.

REST web services – Representational State Transfer (REST) is a style of software architecture for distributed hypermedia systems such as the World Wide Web.

RFC – Remote Function Call. You can use the RFC interface to write applications that communicate with SAP R/3 applications and databases. An RFC is a standalone function. Developers use SAP tools to write the Advanced Business Application Programming (ABAP) code that implements the logic of a function, and then mark it as "remotely callable," which turns an ABAP function into an RFC.

role – Roles control access to Sybase Unwired Platform resources. *See also* logical role and physical role.

role mapping – Maps a physical (server role) to a logical (Unwired Platform role). Role mappings can be defined by developers, when they deploy an MBO package to a development Unwired Server, or by platform or domain administrators when they assign a security configuration to a domain or deploy a package to a production Unwired Server (and thereby override the domain-wide settings in the security configuration).

RSOE – Relay Server Outbound Enabler. An RSOE is an application that manages communication between Unwired Server and a relay server.

runtime server – An instance of Unwired Server that is running. Typically, a reference to the runtime server implies a connection to it.

SAP – SAP is one of the EIS types that Unwired Platform supports.

SCC – Sybase Control Center. A Web-based interface that allows you to administer your installed Sybase products.

schedule – The definition of a task (such as the collection of a set of statistics) and the time interval at which the task must execute in Sybase Control Center.

scheduled data refresh – Data is updated in the cache database from a back-end EIS, based on a scheduled data refresh. Typically, data is retrieved from an EIS (for example, SAP) when a device user synchronizes. However, if an administrator wants the data to be preloaded for a mobile business object, a data refresh can be scheduled so that data is saved locally in a cache. By preloading data with a scheduled refresh, the data is available in the information server when a user synchronizes data from a device. Scheduled data refresh requires that an administrator define a cache group as "scheduled" (as opposed to "on-demand").

security configuration – Part of the application user and administration user security. A security configuration determines the scope of user identity, authentication and authorization checks, and can be assigned to one or more domains by the platform administrator in Sybase Control Center. A security configuration contains:

- A set of configured security providers (for example LDAP) to which authentication, authorization, attribution is delegated.
- Role mappings (which can be specified at the domain or package level)

security provider – A security provider and it's repository holds information about the users, security roles, security policies, and credentials used by some to provide security services to Unwired Platform. A security provider is part of a security configuration.

security profile – Part of the Unwired Server runtime component security. A security profile includes encryption metadata to capture certificate alias and the type of authentication used by server components. By using a security profile, the administrator creates a secured port over which components communicate.

server connection – The connection between Unwired WorkSpace and a back-end EIS is called a server connection.

server farm – *See also* cluster. Is the relay server designation for a cluster.

server-initiated synchronization – *See* push synchronization.

SOAP – Simple Object Access Protocol. SOAP is an XML-based protocol that enables applications to exchange information over HTTP. SOAP is used when Unwired Server communicates with a Web service.

solution – In Visual Studio, a solution is the high-level local workspace that contains the projects users create.

Solution Explorer – In Visual Studio, the Solution Explorer pane displays the active projects in a tree view.

SSO – Single sign-on. SSO is a credential-based authentication mechanism.

statistics – In Unwired Platform, the information collected by the monitoring database to determine if your system is running as efficiently as possible. Statistics can be current or historical. Current or historical data can be used to determine system availability or performance. Performance statistics are known as key performance indicators (KPI).

Start Page – In Visual Studio, the Start Page is the first page that displays when you launch the application.

structured data – Structured data can be displayed in a table with columns and labels.

structure object – Defined in Unwired WorkSpace, structures hold complex datatypes, for example, a table input to a SAP operation.

subscription – A subscription defines how data is transferred between a user's mobile device and Unwired Server. Subscriptions are used to notify a device user of data changes, then these updates are pushed to the user's mobile device.

Sybase Control Center – Sybase Control Center is the Flash-based Web application that includes a management framework for multiple Sybase server products, including Unwired Platform. Using the Unwired Platform administration perspective in Sybase Control Center, you can register clusters to manage Unwired Server, manage domains, security configurations, users, devices, connections, as well as monitor the environment. You can also deploy and MBO or workflow packages, as well as register applications and define templates for them. Only use the features and documentation for Unwired Platform. Default features and documentation in Sybase Control Center do not always apply to the Unwired Platform use case.

Sybase Control Center *X.X* Service – Provides runtime services to manage, monitor, and control distributed Sybase resources. The service must be running for Sybase Control Center to run. Previously called Sybase Unified Agent.

Sybase Hosted Relay Service – The Sybase Hosted Relay Service is a Web-hosted relay server that enables you to test your Unwired Platform development system.

Sybase Messaging Service – The synchronization service that facilitates communication with device client applications.

Sybase Unwired Platform – Sybase Unwired Platform is a development and administrative platform that enables you to mobilize your enterprise. With Unwired Platform, you can develop mobile business objects in the Unwired WorkSpace development environment, connect to structured and unstructured data sources, develop mobile applications, deploy mobile business objects and applications to Unwired Server, which manages messaging and data services between your data sources and your mobile devices.

Sybase Unwired WorkSpace – Sybase Unwired Platform includes Unwired WorkSpace, which is a development tool for creating mobile business objects and mobile applications.

synchronization – A synchronization method where data is delivered synchronously using an upload/download pattern. For push-enabled clients, synchronization uses a "poke-pull" model, where a notification is pushed to the device (poke), and the device fetches the content (pull), and is assumed that the device is not always connected to the network and can operate in a disconnected mode and still be productive. For clients that are not push-enabled, the default synchronization model is pull.*See also* messaging based synchronization.

synchronization group – Defined in Unwired WorkSpace, a synchronization group is a collection of MBOs that are synchronized at the same time.

synchronization parameter – A synchronization parameter is an MBO attribute used to filter and synchronize data between a mobile device and Unwired Server.

synchronization phase – For replication based synchronization packages, the phase can be an upload event (from device to the Unwired Server cache database) or download event (from the cache database to the device).

synchronize – *See also* data refresh. Synchronization is the process by which data consistency and population is achieved between remote disconnected clients and Unwired Server.

truststore – The location in which certificate authority (CA) signing certificates are stored. *See also* keystore.

undeploy – Running **undeploy** removes a domain package from an Unwired Server.

Unwired Server – The application server included with the Sybase Unwired Platform product that manages mobile applications, back-end EIS synchronization, communication, security, transactions, and scheduling.

user – Sybase Control Center displays the mobile-device users who are registered with the server.

view – A window in a perspective that displays information about one or more managed resources. Some views also let you interact with managed resources or with Sybase Control Center itself. For example, the Perspective Resources view lists all the resources managed by the current perspective. Other views allow you to configure alerts, view the topology of a replication environment, and graph performance statistics.

Visual Studio – Microsoft Visual Studio is an integrated development environment product that you can use to develop device applications from generated Unwired WorkSpace code.

Welcome page – In Eclipse, the first set of pages that display when you launch the application.

workspace – In Eclipse, a workspace is the directory on your local machine where Eclipse stores the projects that you create.

WorkSpace Navigator – In Eclipse, the tree view that displays your mobile application projects.

WSDL file – Web Service Definition Language file. The file that describes the Web service interface that allows clients to communicate with the Web service. When you create a Web service connection for a mobile business object, you enter the location of a WSDL file in the URL.

Index