SYBASE®
An SAP® Company

Sybase Control Center for Sybase Unwired Platform

Sybase Unwired Platform 2.0

DOCUMENT ID: DC01092-01-0200-01
LAST REVISED: April 2011

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

Contents

Get Started

Begin using 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

Learn more about Sybase® Unwired Platform documentation.

Table 1. Sybase Unwired Platform Documentation

Document	Description
Sybase Unwired Platform Installation Guide	Describes how to install or upgrade Sybase Unwired Platform. Check the *Sybase Unwired Platform Release Bulletin* for additional information and corrections. Audience: IT installation team, training team, system administrators involved in planning, and any user installing the system. Use: during the planning and installation phase.
Sybase Unwired Platform Release Bulletin	Provides information about known issues, and updates. The document is updated periodically. Audience: IT installation team, training team, system administrators involved in planning, and any user who needs up-to-date information. Use: during the planning and installation phase, and throughout the product life cycle.
New Features	Describes new or updated features. Audience: all users. Use: any time to learn what is available.
Fundamentals	Describes basic mobility concepts and how Sybase Unwired Platform enables you design mobility solutions. Audience: all users. Use: during the planning and installation phase, or any time for reference.

Document	Description
System Administration	Describes how to plan, configure, manage, and monitor Sybase Unwired Platform. Use with the *Sybase Control Center for Sybase Unwired Platform* online documentation. Audience: installation team, test team, system administrators responsible for managing and monitoring Sybase Unwired Platform, and for provisioning device clients. Use: during the installation phase, implementation phase, and for ongoing operation, maintenance, and administration of Sybase Unwired Platform.
Sybase Control Center for Sybase Unwired Platform	Describes how to use the Sybase Control Center administration console to configure, manage and monitor Sybase Unwired Platform. The online documentation is available when you launch the console (**Start** > **Programs** > **Sybase** > **Sybase Control Center**, and select the question mark symbol in the top right quadrant of the screen). Audience: system administrators responsible for managing and monitoring Sybase Unwired Platform, and system administrators responsible for provisioning device clients. Use: for ongoing operation, administration, and maintenance of the system.
Troubleshooting	Provides information for troubleshooting, solving, or reporting problems. Audience: IT staff responsible for keeping Sybase Unwired Platform running, developers, and system administrators. Use: during installation and implementation, development and deployment, and ongoing maintenance.

Document	Description

Tutorials	Tutorials for trying out basic development functionality.
	Audience: new developers, or any interested user.
	Use: after installation.
	• Learn mobile business object (MBO) basics, and create a mobile device application:
	  • *Tutorial: Mobile Business Object Development*
	• Create native mobile device applications:
	  • *Tutorial: BlackBerry Application Development*
	  • *Tutorial: iOS Application Development*
	• Create a mobile workflow package:
	  • *Tutorial: Mobile Workflow Package Development*

Sybase Unwired WorkSpace – Mobile Business Object Development	Online help for developing MBOs. Audience: new and experienced developers. Use: after system installation.
Sybase Unwired WorkSpace – Mobile Workflow Package Development	Online help for developing mobile workflow applications. Audience: new and experienced developers. Use: after system installation.
Developer guides for device application customization	Information for client-side custom coding using the Client Object API. Audience: experienced developers. Use: to custom code client-side applications. • *Developer Guide for BlackBerry* • *Developer Guide for iOS* • *Developer Guide for Mobile Workflow Packages* • *Developer Guide for Windows and Windows Mobile*

Document	Description
Developer guide for Unwired Server side customization – *Developer Guide for Unwired Server*	Information for custom coding using the Server API. Audience: experienced developers. Use: to customize and automate server-side implementations for device applications, and administration, such as data handling. Dependencies: Use with *Fundamentals* and *Sybase Unwired WorkSpace – Mobile Business Object Development*.
Developer guide for system administration customization – *Developer Guide for Unwired Server Management API*	Information for custom coding using administration APIs. Audience: experienced developers. Use: to customize and automate administration at a coding level. Dependencies: Use with *Fundamentals* and *System Administration*.

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.

Administrators

Administrators interact with Unwired Platform primarily to configure it and ensure the production environment works efficiently as a result of that configuration.

Unwired Platform administrators can be one of two types: platform administrators or domain administrators. The platform administrator has cluster-wide administration access to the Unwired Platform, while the domain administrator has access to the specific domains that the cluster-wide administrator assigns. The "SUP Administrator" and "SUP Domain Administrator" logical roles are mapped to specific physical roles in the security repository and used to implement access control and delegate privileges for the platform administrator and domain administration respectively. These roles may or may not be used for protecting mobile business objects as well.

The domain-level role mappings for the aforementioned logical roles in the 'admin' security configuration in the 'default' domain enable the platform adminstrator or domain administrator access control to Unwired Platform. Both types of administrators are authenticated and authorized by the security provider of the 'admin' security configuration. All administrative and application users and their passwords are managed in the security repository. Unwired Platform delegates security checks by passing login and password information to the security provider of the 'admin' security configuration.

Sybase Control Center limits feature visibility depending on the role an administrator logs in with. The platform administrator login has access to the full cluster, whereas the domain administrator login can view information pertaining to only the assigned domains.

"supAdmin" is the default login for cluster-wide administration and "supDomainAdmin" is the default login for domain administration. These logins are assigned "SUP Administrator" and "SUP Domain Administrator" roles, respectively, in the OpenDS LDAP server (Developer Edition). When setting up Deployment Edition security, the administrator specifies physical role mappings for the "SUP Administrator" and "SUP Domain Administrator" logical roles. Once this is configured, only logins with the required physical roles will have access to Unwired Server as either a platform administrator or domain administrator.

Note: The terms Unwired Platform administrator and domain administrator are used in this document to refer to the user with "SUP Administrator" role and "SUP Domain Administrator" role respectively.

Unwired Platform Administrator

An Unwired Platform administrator performs high-level, platform-wide management; by default, the supAdmin login is the Unwired Platform administrator in Developer Edition installation. The Unwired Platform administrator has unlimited access to the Unwired Platform components installed on the network.

To set up a user as Unwired Platform administrator, ensure that the user's login is assigned the physical role mapped to the "SUP Administrator" logical role in the security provider of

'admin' security configuration in the 'default' domain. Once completed, if the login is successfully authenticated and authorized, it is granted the required platform administrator privileges.

Upon First Login

When logging in to the developer edition of Sybase Control Center for the first time, you can use the default ID and password of supAdmin and s3pAdmin, respectively. When logging in to the deployment edition, use the ID and password that are associated with the platform administrator role defined in the "admin" security configuration.

Once authenticated, this platform administrator sees the complete set of tree nodes the role can administer as Unwired Platform managed resources. With these nodes, the administrator can exclusively administer the Unwired Server cluster environment and devices, and delegate domain administration privileges to domain administrators.

Important Considerations

- A user with the Unwired Platform administrator privileges is implicitly granted the domain administrator access for all domains.

Unwired Platform Administration Tasks

Review the tasks an Unwired Platform administrator can perform.

Component	Available tasks
Domain	• View all domains and perform all actions supported within each tenant domain • Create and delete domains
Server	• Configure servers • Perform server starts and stops • View and manage server logs
User	• Access the Unwired Platform administrative users repository • Register domain administrators and assign them to one or more tenant domains • Remove a user from a domain's administrators list
Cluster	• Track the overall health of the cluster, irrespective of domain tenants

Component	Available tasks
Security	• Create named security configurations • Specify authentication, authorization, attribution, and audit providers • Map logical roles to physical roles for security providers • Assign named security configurations to domains or remove preexisting configurations where necessary
Device Registration	• Register devices
Mobile Workflows	• Manage mobile workflows
Monitoring	• Configure monitoring settings and review monitoring data

Domain Administrator

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

To set up a user as a domain administrator, ensure that the user's login is assigned the physical role mapped to the "SUP Domain Administrator" logical role in the security provider of 'admin' security configuration in the 'default' domain. Once completed, if the login is successfully authenticated and authorized, it is granted domain administrator privileges to the domains assigned by Unwired Platform administrator.

When logging in to the developer edition of Sybase Control Center for the first time, domain administrators can use the default ID and password of supDomainAdmin and s3pDomainAdmin, respectively. When logging in to the deployment edition, use the ID and password that are associated with the domain administrator role defined in the "admin" security configuration.

The domain administrator sees a scaled down view of assigned domains in the left navigation pane in Sybase Control Center. This administration role has access only to packages, connections, security configurations, and role mappings in their assigned domain.

Domain Administration Assignments

Only a platform administrator can assign domain administrator access, and only to users that are registered with Unwired Server. Click the Security node in the Sybase Control Center navigation pane to see a list of registered domain administrators.

The number of users assigned this role can vary:

Ratio	Used when
1 administrator:1 domain	Used by small deployments. In this case, the platform administrator is the only administrator. No domain administrator is required.
1 administrator:N domains	Used when multiple domains are required.
N administrators:N domains	Used in most large-scale deployments, or where different domains may be created to support different organizations. In this case, each domain has its own administrator.

Domain Administration Tasks

Review the tasks a domain administrator can perform.

Component	Available tasks
Domain	• Manage the domain • Enable/disable managed domain
Security	• Map logical roles to physical roles for security providers
Log	• View domain log events and settings • Manage package log settings
Packages	• Import, export, deploy, and delete packages • Manage or view the properties of each package, for example perform role mapping, set security configuration, manage subscriptions, view client log, and review MBO/Operation error history. • Enable/disable packages
Connections	• Create and configure server connections • Manage connection templates

Getting Started with Unwired Server Administration

Perform these tasks to access the Unwired Server management console in Sybase Control Center (SCC).

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 Unified Agent in an Unwired Platform enviornment.

- By default, the Sybase Unified Agent 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 Sybase Unified Agent. If the service is running, the status column displays "Started."
  d) To start or stop the agent, right-click the agent 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) (Recommended) Change to the installation directory, %SYBASE%, and run **SYBASE.bat**.
  b) Enter:
  ```
  %SYBASE_UA%\bin\uafstartup.bat
  ```
- Manually stopping Sybase Control Center by command-line script:
  a) (Recommended) Change to the installation directory, %SYBASE%, and run **SYBASE.bat**.
  b) Enter:
  ```
  %SYBASE_UA%\bin\uafshutdown.bat
  ```

Logging in and Starting an Unwired Server Session

Perform these tasks to access the Unwired Platform administration console in Sybase Control Center (SCC).

Consider these typical scenarios:

- When an Unwired Server located on the same host computer as SCC, it is automatically registered by the Unwired Platform installer as a managed resource. Therefore, upon a

successful login, administration console for the Unwired Server opens. Use supAdmin/s3pAdmin as the initial login and password to immediately get started managing and configuring your Unwired Platform environment.
- When SCC is installed with either the Enterprise Developer or the Deployment Editions of Unwired Platform, Unwired Servers in a cluster automatically send discovery requests to detect other Unwired Servers running on the network. When found, they are also automatically registered Sybase Control Center.
- When SCC is installed with a Personal Developer Edition installation, only local server is registered. The discovery properties are configured via Sybase Control Center configuration file. See *System Administration > System Reference > Configuration Files*.

These tasks identify the tasks that get you started with SCC.

Validating Browser Prerequisites

Ensure that your browser meets the appropriate requirements to launch Sybase Control Center.

Sybase Control Center does not support all browsers. Ensure the browser you use to connect to this console is supported. For details, *see Sybase Control Center Requirements* in the *Installation Guide*.

Launching Sybase Control Center for Unwired Platform

Launch the Sybase Control Center administration console remotely or from the desktop to access the login screen.

1. Confirm that the Sybase Unified Agent and Unwired Server services are running.
2. Launch Sybase Control Center using one of:
   - Launch Sybase Control Center from the desktop – double-click the Sybase Control Center shortcut.
   - Connect to Sybase Control Center remotely – open a Web browser and enter `https://<hostname>.<domainname>:<port>/scc`. The default port is 8283. Use this option if you do not have an instance of Sybase Control Center installed on your machine.

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. If you are opening a browser on the same computer as Sybase Control Center, the installer automatically sets up local security certificates.
- 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>\JDK1.6.0_16\bin\keytool.exe -
      exportcert -alias jetty
      -keystore <UnwiredPlatform_InstallDir>\services
      \EmbeddedWebContainer\keystore -file cert.crt
      ```

   b) Click **Start** > **Run**, type `mmc`, and then click **OK** to import the `cert.crt` file into the host computer's Windows store with the Windows Certificate Manager.

Next

You can now open Sybase Control Center from any browser without generating a security exception.

Logging in to Sybase Control Center to Administer Unwired Platform

Log in to Sybase Control Center to access Unwired Platform administration features that you have been authorized to use. Administrators of any Sybase product can log in to Sybase Control Center. However, only users assigned to the SUP Administrator or SUP Domain Administrator roles for Unwired Platform can log in to Unwired Server from Sybase Control Center.

In a production environment, although roles are established, you still need to map Unwired Platform administrator roles to existing roles in the security provider's repository. Perform role mapping after you log in to Sybase Control Center.

When administering a remote server, logging in to Sybase Control Center gains you access only to the Sybase Control Center interface. You must still authenticate later with Unwired Server.

1. In Sybase Control Center, enter a valid:

- **User name** – can be a valid Unwired Platform user that is assigned an administration role. The default Unwired Platform administrator user name is supAdmin.
- **Password** – must be valid for the user name you provide. The password field is case-sensitive. The default supAdmin password is s3pAdmin.

2. Click **Login**.

Note: The administrator account is locked out after three unsuccessful login attempts.

Assigning the SCC User Role to an Unwired Platform User Login

Ensure that the Sybase Control Center sccUserRole role is assigned to an Unwired Platform user login.

Prerequisites

You need to login in as the SCC administrator to perform this role assignment. By default, the supAdmin is assigned the SCC administrator and user role.

Task

1. From the menu bar in SCC, select **Application** > **Administration**.
2. In the left navigation pane of the Sybase Control Center Properties window, expand the **Security** folder.
3. Select **Logins**.
4. In the right administration pane, select the Unwired Platform user login to which you want to assign the SCC user role.

   For information on adding a login to Sybase Control Center, see *Sybase Control Center > Configure > Configuring Sybase Control Center > Authorization > Adding a Login to the System*.
5. Below the properties table, select the **Roles** tab.
6. In the Available Roles for Resource column, select **SCC Service:sccUserRole** and click **Add**.
   The SCC user role appears in the roles list for the Unwired Platform user login.
7. Click **OK**.

Changing Passwords for Unwired Platform Users

If you have set up a development deployment thatruns Unwired Server with the included OpenDS server, all development environment users are authenticated using the credentials in the repository. In this case, all passwords must be changed in the user entry as required.

Either download and install Apache Directory Studio, or use another tool of your choosing. In a production environment, you would need to edit the user entries in your native security repository, by using an appropriate tool.

1. Launch Apache Directory Studio.

2. Add a connection from Apache Directory Studio server to OpenDS LDAP server:
   a) Right-click the Connections view, then click **New Connection**.
   b) In the New Connection wizard, configure the following values:

Page	Property	Value
Network Parameters	**Host**	localhost
	Port	10389 (default port for OpenDS installed by SUP)
	Encryption method	No encryption
Authentication	**Authentication method**	Simple Authentication
	Bind DN	cn=Directory Manager
	Bind password	secret (default bind password)
	Save password	checked

   c) On the Authentication page, click **Check Authentication** to validate the properties you have provided.
   d) If the operation is successful, click **Next** , then click **Fetch Base DNs** (OpenDS server is preconfigured with users such as supAdmin, supDomainAdmin and others), and click **OK**.
   e) Click **Finish**.
   The connection is added and available. You can see the new connection in the Connections view, and user accounts in the LDAP Browser view.
3. Use the Apache Directory Studio LDAP Browser view to modify the `userPassword` property of the user. You will need the current password of the user. The default password for user supAdmin is s3pAdmin, for supDomainAdmin is s3pDomainAdmin, and so on for other default users.

Registering a Cluster as an SCC Managed Resource

Manually register a cluster to make it available as an administrable resource.

Manual registration is necessary if the cluster is not located within your network, and not automatically detected and registered in the Sybase Control Center (SCC) Resource Explorer.

1. In the SCC menu, select **Resource** > **Register**.
2. In the right administration pane of the Resource Registration dialog, enter the **Resource Name** for the cluster you want to register.
3. Select the resource type of the cluster:

   - Unwired Server Cluster – Select this resource type to use an Unwired Server administration environment.

- Afaria® – Choose this resource type when you want to use an Afaria provisioning environment. This makes Afaria available as a manageable resource and allows you to launch the Afaria console from within SCC.

4. (Optional) Enter a description of the cluster.
5. Click **Next**.
6. Enter the **Host Name** and **Port Number** of the primary server in the cluster.

   Depending on the information you entered in the previous steps, one or both of these fields may already be filled out.
7. Click **Finish**.

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**.
   The Unwired Server is added to the Perspective Resources window.
4. In the Perspective Resources window, right-click the cluster you want to manage 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. For more information about configuring a shared authentication repository for both SCC and Unwired Platform, see *Implementing System Wide Security* in the *System Administration* guide.
6. Click **OK**.
7. Right-click the cluster you want to open and select **Manage**.

If you are successfully authenticated, the Unwired Platform console appears. If authentication fails, see *Sybase Control Center Issues*.

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**.

Core Unwired Platform Administration Console Nodes

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 1.5.2.

Node	Purpose
Cluster	View general cluster properties and access the server list for the cluster.
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.
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.
Device Users	Add, view, delete, and edit devices and templates and view and delete application users.

Node	Purpose
Security	Add, view, edit, and delete domain administrators. Add or delete a security configuration. Each security configuration contains one or more security providers for authentication, authorization, attribution, and auditing. Once configured, server configurations can be assigned to domains and then mapped to one or more packages, depending on the requirements for each.
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, devices, and, optionally, queue items.
Monitoring	Create and manage settings for monitoring replication synchronization, messaging synchronization, device notification, data change notification, queue, package, user, and cache activities.

Cluster Administration Overview

The goal of cluster administration is to ensure that clusters and servers work smoothly, and scale over time. Cluster administration is mostly a nonroutine administration task.

By default, the Unwired Platform is installed as a one-node cluster, which may be sufficient for development or test environments. However, production deployments of Unwired Platform are likely to require multiple nodes. See *System Administration > Deployment Architecture Planning*.

Table 2. Cluster administration tasks

Task	Frequency	Accomplished by
Installing the cluster	One-time installation per cluster	Unwired Platform installer
Setting up relay servers	One-time initial installation and configuration; occasionally adding servers to the cluster	Manual installation; manual setup using configuration files.
Suspending and resuming server nodes	On demand, as required	Sybase Control Center
Setting cluster properties, including consolidated database settings, monitoring database setup, and so on	Once, or as cluster changes require	Manual configuration using files and `.BAT` scripts.

Task	Frequency	Accomplished by
Administering the runtime databases	Routine to ensure that the database server is monitored and backed up, that there is sufficient space for Unwired Platform metadata and cached data tables, and that performance is within acceptable limits (performance tuning)	Established processes and command line utilities. Consult with your database administrator. See *System Administration > Operation Maintenance*.
Reviewing licensing information, including total licensed devices and currently used licenses count	Occasional, or as device user registration and deregistration occurs	Sybase Control Center. See *System Administration > Operation Maintenance*.

MBO Package Management Overview

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

Packages contain MBOs that are deployed to Unwired Server to facilitate access to back-end data and transactions from mobile devices. Package types include replication-based synchronization (RBS) packages, messaging-based synchronization (MBS) packages, and SAP® DOE-C packages.

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

Table 3. MBO package management tasks

Task	Package type	Frequency	Accomplish by using
Deploy packages to a development or production Unwired Server	RBS and MBS	Once, unless a new version becomes available	Sybase Control Center for Unwired Platform with the domain-level Packages node
Control user access by assigning security configurations for each package, and mapping roles if fine-grained authorization is enforced through logical roles	RBS and MBS	Once, unless security requirements of the package change	Sybase Control Center for Unwired Platform with the domain-level Packages node

Task	Package type	Frequency	Accomplish by using
Set up the package cache interval and cache refresh schedule (for getting data updated on the Unwired Server from the data source)	RBS and MBS	Once, unless data refreshes need to be tuned	Sybase Control Center for Unwired Platform with the domain-level Packages node
Manage subscriptions (RBS, MBS, and DOE-C), synchronization groups (RBS and MBS), and device notifications (RBS) to customize how updated data in the cache is delivered to the device user	Varies	Periodic, as required	Sybase Control Center for Unwired Platform with the domain-level Packages node
Export or import an MBOpackage	RBS and MBS	On-demand, as required	Sybase Control Center for Unwired Platform with the domain-level Packages node
Review current/historical/performance metrics	All	Routine	Sybase Control Center for Unwired Platform with the Monitor node (available only to administrators)

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 4. Server administration tasks

Task	Frequency	Accomplished by
Installing the server	One-time installation per server	Unwired Platform installer.

Task	Frequency	Accomplished by
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 administration communication • Set up secure synchronization • View consolidated database properties • Configure replication and messaging push notifications • Set transport level security properties within a security profile • Tune server performance	Postinstallation configuration with infrequent tuning as required	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.

Device and User Management Overview

The goal of device user management is to register, activate, and provision devices so that device users can access applications and be tracked and managed by the system. This is particularly important for messaging-based synchronization (MBS) environments, because users cannot access the application unless the device is registered and activated.

Table 5. Device and user management tasks

Task	Frequency	Accomplish by using
Activate and deactivate application devices according to the type of synchronization model used (RBS or MBS)	When a new user of a messaging or mobile workflow package must be added	Sybase Control Center for Unwired Platform with the Device Users node.
Review registered devices and users, delete devices to free licenses, delete users to remove them from the system	As required	Sybase Control Center for Unwired Platform with the Device Users node.
Manage subscriptions	As required	Sybase Control Center for Unwired Platform with the Packages node.
Add and manage device templates	As required	Sybase Control Center for Unwired Platform with the Device Users node.

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

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:

- Replication-based synchronization
- Messaging-based synchronization
- System queue status

- Data change notifications
- Device notifications (RBS)
- Package statistics
- Device users
- 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 6. 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

Mobile Workflow Package Administration Overview

The goal of mobile workflow package management is to make mobile workflows available from the Unwired Server to device users. Mobile workflow package management typically requires a one-time deployment and configuration, except for ongoing package maintenance.

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

Table 7. Mobile workflow package management

Task	Frequency	Accomplish by using
Deploy mobile workflow packages	Once, unless a new version becomes available	Sybase Control Center for Unwired Platform with the Workflow node
Mobile workflow configuration that includes e-mail matching rules and context variables	Once	Sybase Control Center for Unwired Platform with the Workflow node
Device registration and user assignments to mobile workflow packages	Routine when new users or new devices are added	Sybase Control Center for Unwired Platform with the **Workflow** > <**WorkflowName**> node
Monitor users and errors	Routine	Sybase Control Center for Unwired Platform with the Monitor node

Configure

Configure your environment to use Sybase Control Center to monitor and manage your resources.

Configuring Unwired Platform

Use Sybase Control Center for Unwired Platform to 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.

These elements are listed here.

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.

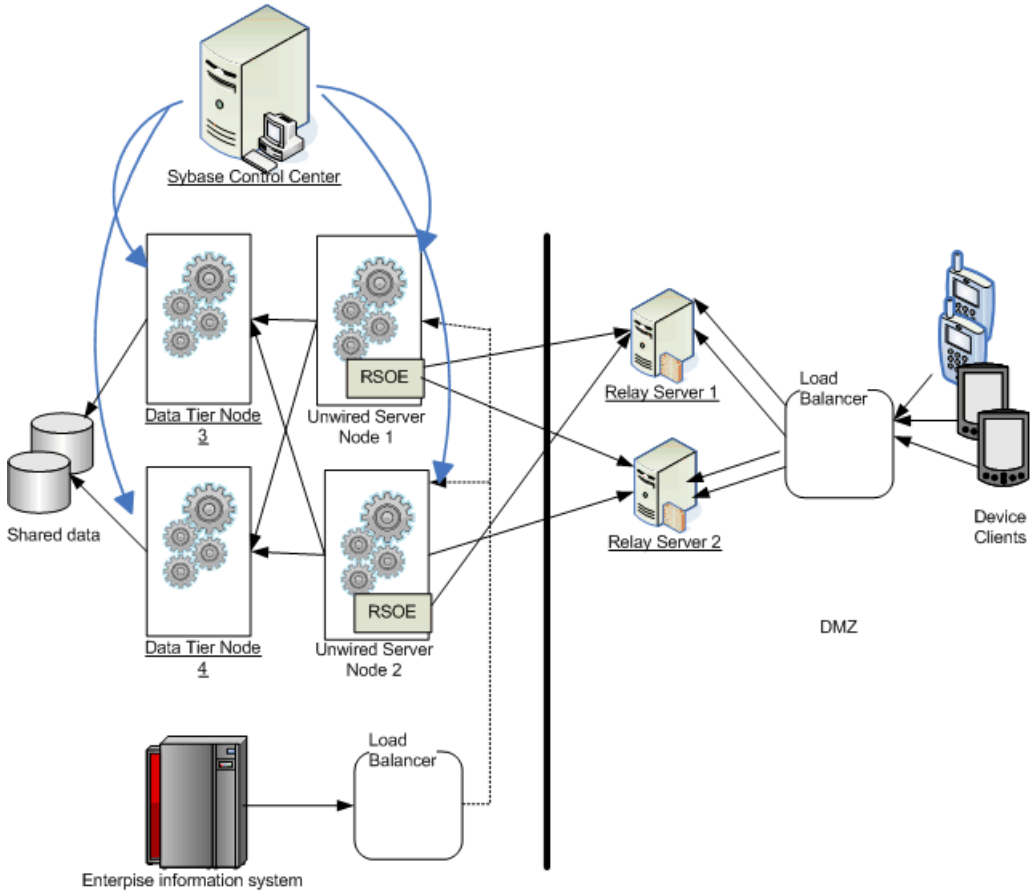
Tables that support copying and pasting include monitoring properties, device properties, user properties, registration templates, domain log properties, and sever log properties.

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).

Relay Server

The relay server (rshost.exe), is a required component in a highly-available production environment, and adds an extra layer of security and load balancing to the server environment. This server component, which is deployed into the enterprise demilitarized zone (DMZ), allows secure communication between devices and Unwired Server components across the firewall.

The relay server enables an outbound connection model. The Unwired Platform components make outbound connections into the enterprise DMZ using the HTTPS protocol. When using a relay server, connections from all device clients end within the DMZ of your enterprise:



The relay server also supports load balancing in your Unwired Platform installation by accepting requests from remote clients, and forwarding the requests to a farm of Unwired Servers. If you are using Afaria, you can configure your Afaria environment to share one relay server.

Sybase offers two relay server options:

- For development and testing – the Sybase-hosted relay service, an Internet-hosted relay server that supports an Unwired Server cluster with a two-server limitation.
- For production – the Sybase-installed relay server, which supports Unwired Server with an unlimited number of redundant servers.

The relay server is implemented as a Web extension or a plug-in that runs in a Web server. Unwired Platform supports two Web servers: IIS on Windows, and Apache on Linux. The relay server accepts device client requests and distributes them across a farm of Unwired Servers and Afaria servers. Each individual Unwired Server, Unwired Server embedded component, and Afaria server must run a relay server outbound enabler (RSOE), which

establishes a permanent connection to the relay server on behalf of the server or embedded server component.

Configuring a Relay Server for a Cluster

Choose a method of configuring a new relay server. After you generate a configuration, you can copy it to the relay server host to quickly distribute the same setup to multiple relay server nodes with minimal changes.

This task applies only to an installed instance of relay server in production environments. It does not apply to development or test environments that use a Sybase-hosted relay server.

1. Configure a relay server.
   - Perform a quick configuration to use defaults for all but environment-specific properties.
   - Perform a new configure to configure all properties of the relay server.
2. Generate the configuration file.
3. Transfer the file to all relay server hosts on your network, making any required modifications.

Creating a Quick Configuration

Create a relay server configuration primarily with system defaults and generate required RSOE processes for each server node detected.

1. In the navigation pane, click the cluster name.
2. In the administration pane, click the **Relay Servers** tab.
3. Click **Quick Configure**.
4. Configure these relay server properties and click **OK**:
   - **Host –** either the relay server host name or the host name of the load balancer (if one is used in your environment). Device clients use the host name to establish connections. Other relay servers can use the host name to identify peers if a load balancer is not used.
   - **HTTP port –** the HTTP port for relay server connections.
   - **HTTPs port –** the secure HTTP port for relay server connections.
   - **(Optional) Description –** description for the relay server.
   - **(Optional) URL suffix –** A URL suffix that allows a device application client to connect to a relay server farm. The default value for Apache on Linux is `/srv/iarelayserver`, and the default value for IIS on Windows is `/ias_relay_server/server/rs_server.dll`.

Farm and server node properties are automatically populated based on the Unwired Server host computer name. Tokens are autogenerated. Six RSOE instances (three each for MBS and RBS services) on each server node are created, but not started.

Note: Developers need these values to configure correct property values for Unwired Server connections. Values in both places must exactly match, or the connection to the Unwired Server via the relay server fails.

Next

Review and edit the values in the relay server configuration; for example, change the Token value to a custom one for your deployment.

Creating a Custom Relay Server Configuration

Create a relay server configuration by manually configuring all relay server properties. Upon completion, set up one relay server outbound enabler (RSOE) for each server node.

Launching the Relay Server Configuration Wizard

Launch the Relay Server Configuration wizard to create a new relay server configuration file with custom values.

1. In the navigation pane, click the 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. Configure these relay server properties:
   - **Host –** either the relay server host name or the host name of the load balancer (if one is used in your environment). Device clients use the host name to establish connections. Other relay servers can use the host name to identify peers if a load balancer is not used.
   - **HTTP port –** the HTTP port for relay server connections.
   - **HTTPs port –** the secure HTTP port for relay server connections.
   - **(Optional) URL suffix –** A URL suffix that allows a device application client to connect to a relay server farm. The default value for Apache on Linux is `/srv/iarelayserver`, and the default value for IIS on Windows is `/ias_relay_server/server/rs_server.dll`.
   - **(Optional) Description –** description for the relay server.

Note: Developers need these values to configure correct property values for Unwired Server connections. Values in both places must exactly match, or the connection to the Unwired Server via the relay server fails.

2. Click **Next**.

Define Server Farms and Cluster Nodes

Add one or more relay server farms to distribute the workload as needed. A relay server farm expedites computing processes by harnessing the power of multiple back-end server nodes in a particular Unwired Platform cluster. Remove farms or nodes as required.

Prerequisites

Determine environment type and required degree of redundancy:

- In a production environment, add multiple relay server farms when a load balancer is used to distribute synchronization requests.
- In a development testing environment, configure a single farm with a single relay server to more closely replicate production values.

Task

Repeat these steps to add or remove multiple farms as needed.

1. Create the relay server farm:
   a) Configure these properties:
      - **Farm ID –** the server farm for which the relay server manages requests. This property is case-sensitive. The configured value must match the value defined for the RSOE or the connection fails.
      - **Type –** the type of request managed by the relay server: replication based synchronization (RBS) or messaging based synchronization (MBS).
      - **(Optional) Description –** describes the relay server farm usage.
   b) Click +.
   c) Repeat steps 1 and 2 to add multiple server farms.
   d) To delete a configured farm, select it in the list of configured farms, then click the **X** button.
2. Assign the farm to a server node for the type you created:
   a) Select an existing farm name you want to assign a server node to:
   b) Configure these properties:
      - **Node ID –** the node string that identifies the backend replication or messaging based cluster. Combine one or more Unwired Servers to create a single server node. This property is case-sensitive. The configured value must match the value defined for the RSOE or the connection fails.

- **Token –** the security token used by the server node to authenticate the back-end server connection with relay server. Each node requires a unique token: specify a unique string to a maximum of 2048 characters.

c) Click +.
d) Repeat steps 1 and 2 to add multiple server nodes.
e) To delete a configured node, select it in the list of configured farms and click **X**.

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

Reviewing Configured Relay Server Properties

Review the relay server configuration to validate properties, before generating its configuration file.

1. Review the properties, ensuring that:
   - No errors exist.
   - All farms are defined and are of the correct type.
2. Click **Finish**.

The relay server is registered with Sybase Control Center, and can be managed from the **Relay Servers** tab for the cluster it was added for.

Next

When you have finished adding all required relay servers, set up one or more relay server outbound enabler (RSOE) for each server node you define with this method. Add RSOE processes to each Unwired Server node listed in the Servers folder of the Sybase Control Center navigation pane.

Generating the Relay Server Outbound Enabler Configuration File

To quickly and easily replicate a standard configuration to multiple hosts, generate a `rsoeconfig.xml` file.

An administrator might use Sybase Control Center to configure an initial RSOE for development. Once a configuration proves valid and stable, the administrator can generate this file before using `regRelayServer.bat` to apply it to other server nodes .

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

Generating the Relay Server Configuration File

Generate all or part of the `rs.config` file used to configure a relay server. Then transfer the generated output to all required relay server hosts.

Generating a file extracts values stored in the cluster database during the configuration process, and writes them to a file.

1. In the navigation pane, click the 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 file
   - Either the server node definition area or the farm definition area
6. Select an output target for the file.
7. Click **Finish**.

Managing Configured Relay Servers

Relay servers configured with Sybase Control Center are registered in the cluster database. Administrators can view or edit 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 custom values.

1. In the navigation pane, click the 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. Configure these relay server properties:
   - **Host –** either the relay server host name or the host name of the load balancer (if one is used in your environment). Device clients use the host name to establish connections. Other relay servers can use the host name to identify peers if a load balancer is not used.
   - **HTTP port –** the HTTP port for relay server connections.
   - **HTTPs port –** the secure HTTP port for relay server connections.
   - **(Optional) URL suffix –** A URL suffix that allows a device application client to connect to a relay server farm. The default value for Apache on Linux is `/srv/iarelayserver`, and the default value for IIS on Windows is `/ias_relay_server/server/rs_server.dll`.
   - **(Optional) Description –** description for the relay server.

   Note: Developers need these values to configure correct property values for Unwired Server connections. Values in both places must exactly match, or the connection to the Unwired Server via the relay server fails.
2. Click **Next**.

Define Server Farms and Cluster Nodes

Add one or more relay server farms to distribute the workload as needed. A relay server farm expedites computing processes by harnessing the power of multiple back-end server nodes in a particular Unwired Platform cluster. Remove farms or nodes as required.

Prerequisites

Determine environment type and required degree of redundancy:

- In a production environment, add multiple relay server farms when a load balancer is used to distribute synchronization requests.
- In a development testing environment, configure a single farm with a single relay server to more closely replicate production values.

Task

Repeat these steps to add or remove multiple farms as needed.

1. Create the relay server farm:
   a) Configure these properties:
      - **Farm ID –** the server farm for which the relay server manages requests. This property is case-sensitive. The configured value must match the value defined for the RSOE or the connection fails.

- **Type –** the type of request managed by the relay server: replication based synchronization (RBS) or messaging based synchronization (MBS).
- **(Optional) Description –** describes the relay server farm usage.

b) Click +.
c) Repeat steps 1 and 2 to add multiple server farms.
d) To delete a configured farm, select it in the list of configured farms, then click the **X** button.

2. Assign the farm to a server node for the type you created:

a) Select an existing farm name you want to assign a server node to:
b) Configure these properties:

- **Node ID –** the node string that identifies the backend replication or messaging based cluster. Combine one or more Unwired Servers to create a single server node. This property is case-sensitive. The configured value must match the value defined for the RSOE or the connection fails.
- **Token –** the security token used by the server node to authenticate the back-end server connection with relay server. Each node requires a unique token: specify a unique string to a maximum of 2048 characters.

c) Click +.
d) Repeat steps 1 and 2 to add multiple server nodes.
e) To delete a configured node, select it in the list of configured farms and click **X**.

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

Reviewing Configured Relay Server Properties

Review the relay server configuration to validate properties, before generating its configuration file.

1. Review the properties, ensuring that:

- No errors exist.
- All farms are defined and are of the correct type.

2. Click **Finish**.

The relay server is registered with Sybase Control Center, and can be managed from the **Relay Servers** tab for the cluster it was added for.

Next

When you have finished adding all required relay servers, set up one or more relay server outbound enabler (RSOE) for each server node you define with this method. Add RSOE processes to each Unwired Server node listed in the Servers folder of the Sybase Control Center navigation pane.

Deleting a Relay Server Configuration

Delete a relay server configuration to remove all defined farms, server nodes, and RSOEs that connect to this relay server.

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

Relay Server Tab Data Reference

The columns of data that appear in the Relay Server tab for an Unwired Platform cluster node.

Column	Description
Host	either the relay server host name or the host name of the load balancer (if one is used in your environment). Device clients use the host name to establish connections. Other relay servers can use the host name to identify peers if a load balancer is not used.
HTTP Port	the HTTP port for relay server connections.
HTTPS Port	the secure HTTP port for relay server connections.
URL Suffix	A URL suffix that allows a device application client to connect to a relay server farm. The default value for Apache on Linux is `/srv/iarelayserver`, and the default value for IIS on Windows is `/ias_relay_server/server/rs_server.dll`.

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 consolidated 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 sychronization 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. See *Sybase Control Center online help > Get Started > Getting Started with Unwired Platform Administration > Getting Started with Remote Server Administration > Registering a Remote Cluster as an SCC Managed Resource*.

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.

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 Ports

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

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. See the next topic, *Security Key and Certificate Basics*.

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.

Management Ports

Management ports in Unwired Server process incoming administration connection requests and manage these requests.

To configure the administration listener, you must know whether you want the communication processes secured (using SSL) or unsecured. If you choose an unsecured administration process stream, simply configure the port. For secured administration processes, choose the protocol and port, and also perform other setup tasks that complete the implementation so that the stream is completely encrypted.

For information on complete setup tasks required to enable SSL in the server administration environment, see *System Administration for Sybase Unwired Platform > Security Administration > Implementing System Wide Security > Transport Security Setup > Encrypting Unwired Server Administration Connections*.

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 Unwired Server Administration Connections* in the *System Administration* guide.

Task

The following taks are involved in enabling and configuring SSL encryption:

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.

Task

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.

6. Click **Save**.

Next

Create an SSL security profile that uses the selected certificates.

Creating an SSL Security Profile

Create a security profile that defines the certificate alias and authentication levels used to encrypt communication ports in Unwired Platform.

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

Profile	Authenticates	Cipher suites
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	• 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
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

Ensure that SSL is enabled for every node in the cluster, then restart all servers in the cluster to commit the configuration changes. Log out of Sybase Control Center, then log back in on the secure port. For information on setup tasks required to enable SSL in the server administration environment, see *System Administration for Sybase Unwired Platform > Security Administration > Implementing System Wide Security > Transport Security Setup > Encrypting Unwired Server Administration Connections*.

Consolidated Database

The consolidated database (CDB) is a runtime cache database used by Unwired Server.

The CDB is a required component of Unwired Platform. By default, an embedded SQL Anywhere® database server is used as the CDB. However, during installation you can configure Unwired Server to use an existing SQL Anywhere instance as its CDB server.

Administration considerations

If you install multiple Unwired Servers in a load-balancing cluster, all Unwired Servers in the same cluster must share a CDB; however, in this scenario, a CDB failure can introduce a single point of failure for Unwired Platform. To mitigate this risk, you can run the CDB in failover mode using a shared-disk cluster. For information about implementing server and data tier clusters with optimal redundancy, see *Deployment Planning* in the *System Administration* guide.

Depending on you environment, the location of the CDB database file varies:

- In cluster environment, the default file location is `<UnwiredPlatform_InstallDir>\Data\CDB\default.db`.
- For all other environments, the default file location is `<UnwiredPlatform_InstallDir>\Servers\UnwiredServer\data\default.db`.

If you choose an existing database instance as your CDB server, ensure that the database is configured to be case-insensitive; otherwise, issues with primary keys may arise during operation of Unwired Platform. In Unwired Platform, consolidated databases and client databases are configured as case insensitive by default. This configuration requires that primary keys in the datasource also be case insensitive. Otherwise, if there are two records with primary keys that just have case differences in the back-end system, they are treated as one record when loading them into the consolidated database. In this case, the second primary key overwrites the first that may exist in the consolidated database already.

Runtime data in the CDB

Besides acting as the synchronization cache for mobile business object (MBO) data, the CDB also contains metadata and runtime data, including:

- Unwired Server properties
- Push subscriptions and status
- Synchronization timestamps for each device client
- User data, such as personalization keys and device tracking information

Viewing Consolidated Database Properties

Review the consolidated database (CDB) properties that allow Unwired Server to connect to the database.

You cannot use Sybase Control Center to configure CDB properties. .

1. In the left navigation pane, expand the **Servers** folder and select a server.
2. Select **Server Configuration**.
3. In the right administration pane, click the **Consolidated DB** tab.
4. Review these properties:
   - **Database Thread Count** – the number of worker threads used for the CDB. The default value is 20 threads. However, if you are experiencing performance issues, especially in a clustered environment, you may need to increase this value.

     For a SQL Anywhere CDB only, use this formula to estimate a new value:

     ```
     Value = Number of nodes in cluster * (sync threadcount +
     1) + Number of scheduled EIS fetches + 10
     ```

     For example, if you have been using the default synchronization thread count of 20, but have added three Unwired Servers to your cluster, adjust the CDB thread count to 78 or (3 * (20 +1) + 5 + 10). If you set the value for this thread count to 78 or higher, the value is accepted. However, if you set the value lower than 78, the request is ignored, unless you remove some servers from the cluster or reduce the synchronization thread counts.
   - **Database Type** – the CDB type; Sybase_ASA for the default SQL Anywhere database.
   - **Database DSN Name** – CDB DSN name descriptor.
   - **Database Name** – database name descriptor.
   - **Database Server Port** – the port over which CDB communication takes place. The default is 5200.
   - **Database Password** – the database user password. In SQL Anywhere, the default is `sql`.
   - **Database Server Host** – the name of the machine where the existing database server is running.
   - **Database Server** – the name of the database server used to manage requests for the consolidated database. By default, the server is localhost. However, if the database is on another host or is part of a cluster, you may need to use another host name.
   - **Database User** – the CDB user name. In SQL Anywhere, the default user name is `dba`.

5. (Optional) Expand the **Show Optional Properties** section and review values for these properties:
   - **Database Install Type** – the type of database installation; either default or custom.
   - **Database ASA Mode** – the database mode. The value of the first node of a cluster is "primary," the value of the second node is "arbiter," and the value of the third node is "mirror."
   - **Database User Options** – specify the CDB start-up user options.

Replication

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

For replication-based synchronization, configure the corresponding port to receive incoming synchronization requests from devices, as well as set up configuration to enable push notification messages to the device when data changes in CDB.

Configuring a Synchronization Listener for Replication-Based Synchronization

Configure the port to receive replication-based synchronization requests from client devices, and if you are using push synchronization, then also configure synchronization listener properties.

Prerequisites

Determine whether you require an encrypted (secured) or unencrypted synchronization stream. A secure synchronization stream uses SSL encryption; therefore, before setting up a secure configuration, ensure that you possess digital certificates verified and signed by third-party trusted authorities. The HTTPS protocol is slower than the HTTP protocol; use SSL only if you require HTTPS. See *Transport Security Setup* in the *System Administration* guide.

Task

1. In the left navigation pane, expand the **Servers** folder and select the server you want to configure.
2. Select **Server Configuration**.
3. In the right administration pane, click the **Replication** tab.
4. If push synchronization is being added to replication-based synchronization application, select **Synchronization Listener** from the menu bar.
5. Select the protocol and port you require:

- If you do not require SSL encryption, choose **Synchronization port**. Sybase recommends this option if you do not require a secure communication stream for synchronization. By default, the port for HTTP is 2480.
- To encrypt the HTTP stream with SSL, choose **Secure Synchronization port**. By default, the port for HTTPS is 2481.

6. Configure these properties:

   - **Synchronization Cache Size** – sets the maximum cache size for the synchronization port. The default is 50MB.
   - **Thread Count** – sets the number of worker threads used for synchronization. The default is 5. If you experience performance issues, you may need to increase this value.

7. (Optional) Expand the optional properties section to configure these properties:

   Note: You can only configure the encryption properties on the primary Unwired Server. Secondary servers will inherit the values, where they become read-only.

   - Certificate Password – is used to decrypt the private certificate listed in certificate file. You specify this password when you create the server certificate.
   - Certificate – identifies the location of the security certificate used to encrypt and decrypt data transferred using SSL.
   - E2E Encryption Type – specify the asymmetric cipher used for key exchange for end-to-end encryption. You can only use RSA encryption.
   - E2E Encryption Certificate – specify the file containing the private key that acts as the identity file for Unwired Server.
   - E2E Encryption Certificate Password – set the password to unlock the encryption certificate.
   - User Options – sets the command line options for starting the synchronization server. These options are appended the next time the synchronization server starts. These are the available user options:

Option	Description
`@ [variable | filePath]`	Applies listener options from the specified environment variable or text file.
`-a <value>`	Specifies a single library option for a listening library.
`-d <filePath>`	Specifies a listening library.
`-e <deviceName>`	Specifies the device name.
`-f <string>`	Specifies extra information about the device.
`-gi <seconds>`	Specifies the IP tracker polling interval.
`-i <seconds>`	Specifies the polling interval for SMTP connections.

Option	Description
`-l <"keyword=value;...">`	Defines and creates a message handler.
`-m`	Turns on message logging.
`-ni`	Disables IP tracking.
`-ns`	Disables SMS listening.
`-nu`	Disables UDP listening.
`-o <filePath>`	Logs output to a file. **Note:** Ensure that you enter the absolute file path for this property.
`-os <bytes>`	Specifies the maximum size of the log file.
`-ot <filePath>`	Truncates a file, then logs output to that file.
`-p`	Allows the device to shut down automatically when idle.
`-pc [+ | -]`	Enables or disables persistent connections.
`-r <filePath>`	Identifies a remote database involved in the responding action of a message filter.
`-sv <scriptVersion>`	Specifies a script version used for authentication.
`-t [+ | -] <name>`	Registers or unregisters the remote ID for a remote database.
`-u <userName>`	Specifies a synchronization server user name.
`-v [0 | 1 | 2 | 3]`	Specifies the verbosity level for the messaging log.
`-y <newPassword>`	Specifies a new synchronization server password.

Do not use the User Options property in Sybase Control Center to pass in these options: `-c`, `-lsc`, `-q`, `-w`, `-x`, `-zs`.

For more information on synchronization server command line options, see *Listener options for Windows* in the *SQL Anywhere 11.0.1* online help.

8. Click **Save**.

Enabling Push and Pull Notifications

Configure push and pull for replication-based synchronization notifications.

Prerequisites

Determine the type of push synchronization gateway you require.

Task

This task sets the lightweight polling notification configuration for all clients. You can also configure it individually for each device.

1. In the left navigation pane, expand the **Servers** folder and select the server to configure.
2. Select **Server Configuration**.
3. In the right administration pane, click the **Replication** tab.
4. Select **Notification Configuration** from the menu bar.
5. To enable server-initiated push notification:
   a) Select **Enable Push-Based Notification**.

      Unselecting this option disables push synchronization and all related configuration properties.
   b) Enter a time interval in seconds, minutes, or hours to specify the frequency with which the push notification table is polled. The default is 10 seconds.
6. To enable client-initiated push notification:
   a) Select **Enable Pull-Based Notification**.

      Unselecting this option disables pull synchronization and all related configuration properties.
   b) Enter a time interval in seconds, minutes, or hours to specify the frequency with which the pull notification table is polled. The default is 10 seconds.
7. In the Notification Configuration menu, click **Save**.

Messaging

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

To implement messaging synchronization, you must determine the subscription strategy you want to use and how it will support the data change notification strategy that is implemented by the developer.

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

Configuring Messaging-Based Synchronization Properties

Configure one or more synchronization ports to receive service requests from devices.

1. In the left navigation pane, expand the **Servers** folder and select a server.
2. Select **Server Configuration**.
3. In the right administration pane, click the **Messaging** tab, and select **Synchronization Listener**.
4. Enter the synchronization port number. The default is 5001.
5. (Optional) Select **Listen on multiple synchronization ports** and enter the additional port numbers.

   Depending on your environment, listening on multiple synchronization ports may provide greater flexibility and reliability. High activity on particular ports, such as virus detection and data inspection, may result in dropped packets or connections if alternate ports are unavailable. When multiple ports are configured, all messaging traffic is still funneled to a single listener.
6. Click **Save**.

Configuring Apple Push Settings

Create a new Apple Push Notification Service (APNS) configuration that specifies the application, security certificate, and ports that the service uses.

Apple push notifications use SMS-based push to alert offline iPhone users to the availability of new items awaiting retrieval on Unwired Server. SMS-based push uses an IP connection only long enough for the Send/Receive data exchange to complete. The feature overcomes network issues with always-on connectivity and battery life consumption on 3G networks.

For more information on end-to-end iPhone application development and provisioning, see *System Administration for Sybase Unwired Platform > Systems Administration > Device and User Management > Device Provisioning > Apple Provisioning for iPhone*.

Note: When configuring the Apple Push Notification Service, change the push gateway, push gateway port, feedback gateway, and feedback gateway port values only when configuring notifications in a development environment. To enable Apple push notifications, the firewall must allow outbound connections to Apple push notification servers on default ports 2195 and 2196.

1. In the left navigation pane, expand the **Servers** folder and select a server.
2. Select **Server Configuration**.
3. In the Messaging tab, select **Apple Push Configuration**.
4. Click **New**.
5. Enter the **Application name**. This name corresponds to the Product Name specified in Xcode.

6. Select one of:
   - **Use existing certificate** – use a security certificate file that already exists on the server. When you select this option, the list of available certificates appears in the **Certificate name** menu.
   - **Use new certificate** – create a new security certificate. When you select this option, you must provide information to create a named security certificate file on the server.
7. If you selected **Use existing certificate**:
   a) Select the desired certificate from the list.
   b) Enter and confirm the certificate password.
8. If you selected **Use new certificate**:
   a) Enter a name for the new certificate.
   b) Specify a Base64-encoded string by choosing one of these:
      - **Browse from file** – select a security certificate file on the server that contains the Base64-encoded string.
      - **Base64-encoded string** – manually enter the Base64-encoded string.
   c) If you selected a file from the server for the Base64-encoded string, you can overwrite the existing certificate file with the details you specify during new certificate creation. To do so, select the box adjacent to **Overwrite existing certificate**.
   d) Enter and confirm the certificate password.
9. Click **OK**.

Configuring BlackBerry Push Settings

Create a new BlackBerry Push Service configuration that specifies the delivery method for the service users.

BlackBerry push notifications use SMS-based push to alert offline users to the availability of new items awaiting retrieval on Unwired Server. SMS-based push uses an IP connection only long enough for the Send/Receive data exchange to complete. The feature overcomes network issues with always-on connectivity and battery life consumption on 3G networks.

1. In the left navigation pane, expand the **Servers** folder and select a server.
2. Select **Server Configuration**.
3. In the Messaging tab, select **BlackBerry Push Configuration**.
4. Click **New**.
5. Enter the **Name** of the configuration.
6. Enter the **URL** of the delivery service host.
7. (Optional) Enter a **User name** and **Password** required to access the URL. Confirm the password.

8. Select BES delivery. BlackBerry Enterprise Server uses an e-mail service provided by a server ran by a company to deliver messages. If using BES, you can also enable the Push Access Protocol (PAP) if you are using a Push Proxy Gateway to deliver messages.
9. Click **OK**.

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 properties, as required:
   - Host Name – the name of the machine where Unwired Server is running.
   - Thread Stack Size – the JVM `-Xss` option.
   - Minimum Heap Size – the minimum size of the JVM memory allocation pool, in megabytes. Sybase recommends that this value not fall 500 megabytes for a 32-bit operating system, but 1 gigabyte is recommended. For a 64-bit operating system, Sybase recommends 1 gigabyte for a normal configuration, but 2 gigabyte for a stress configuration (which can vary depending on what RAM is available).
   - Maximum Heap Size – the maximum size of the JVM memory allocation pool, in megabytes. For a 32-bit operating system, Sybase recommends a 1.5 gigabyte maximum heap size value. For a 64-bit operating system, Sybase recommends 1 gigabyte for a normal configuration, but 4 gigabyte for a stress configuration (which can vary depending on what RAM is available).

   Note: The synchronization differencing algorithms are a key feature of RBS; this technology runs in the JVM. You must provide adequate memory to these components. If these algorithms are memory starved, the JVM spends an inordinate amount of time garbage collecting memory, and synchronizations back up in the internal queues. You can monitor process memory usage with tools like SysInternal's Process Explorer to determine the actual amount of memory in use by Unwired Platform, and adjust the JVM heap size accordingly

   Note: Always leave 4 gigabytes for the running of the OS and other applications that may exist on the server.
6. (Optional) Expand the **Show optional properties** section and configure these properties, as required:
   - User Options – other JVM options. For example, you can enable JVM garbage collection logging by setting `-XX:+PrintGCDetails`.

- 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**.
8. If your server is installed as a Windows service:
   a) Stop Unwired Server.
   b) Open a command prompt.
   c) Run `sup-server-service.bat remove`.
   d) Run `sup-server-service.bat install auto`.
   e) Restart Unwired Server.

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.

Server Log

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 settings by accessing the Server Configuration node in the left navigation pane. You can also view consolidated database properties.

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.

Configuring Server Log Settings

Configure 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.

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 left navigation pane, expand the **Servers** folder and select the server to configure.
2. Select **Log**.
3. In the right administration pane, click the **Settings** tab.
4. 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.
5. For each of the listed components, choose one of these log levels:

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

The default log levels are:

Component	Default Log Level
MMS	Info
MSG	Info
Security	Info
Mobilink	Info
DataServices	Info

Component	Default Log Level
Other	Warn

6. 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.

Relay Server Outbound Enabler

The relay server outbound enabler (RSOE) runs as an Unwired Server process and manages synchronization requests from a relay server.

Connect, transfer, and disconnect activities of the RSOE are driven on demand by the clients and backend servers. The connections the RSOE facilitates between relay and backend servers are bidirectional. That is, the RSOE, after retrieving information regarding the available relay servers, creates the up and down channel pairs with each relay server:

- Up channel – the relay server forwards client requests to the RSOE using the established up channel (either HTTP or HTTPS). The outbound enabler then relays the client request to the backend Unwired Server component (replication based synchronization or messaging based synchronization).
- Down channel – once the outbound enabler receives the response from the Unwired Server, it forwards the response back to the relay server. The relay server relays the response to the client.

As an Unwired Server process, RSOE always starts when Unwired Server starts. Unwired Server monitors the process to ensure it is available -- if for any reason an RSOE fails, Unwired Server restarts it automatically. At this time, the RSOE must be manually restarted.

Note: Sybase recommends that three RSOE processes be added to each synchronization port (RBS and MBS types) you define as part of an Unwired Server farm for a relay server.

Loading and Unloading HTTPS Certificates for RSOE

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

Prerequisites

You can only use RSA certificates. Do not use ECC certificates, certificates from a third-party source such as openssl, or a createcert-produced certificate from another Sybase product installation.

Task

If the Web server 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 **RSOE** 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 **Overwrite the certificate file**, then click +.
   - To delete a certificate from the store, select the filename and click **X**.
4. When all certificate management tasks are completed, click **OK**.

Setting Up RSOE

Set up one RSOE for each server node defined as part of the relay server configuration. The configured values are saved in the cluster database.

Configuring RSOE General Properties

Configure general properties for RSOE. These properties 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 **RSOE** tab, then click **New**.
3. Configure these properties:
   - **Farm type –** the type of request managed by the relay server: replication based synchronization (RBS) or messaging based synchronization (MBS).
   - **Unwired sever port –** select the port on which RSOE will manage synchronization requests. The default ports are 2480 for RBS, 2481 for secure RBS, and 5001 for MBS.

- **Relay server host –** type the relay server host RSOE connects to.
- **Relay server port –** select the relay server port.
- **Unwired server farm –** select the name of the server farm.
- **Server node ID –** select the ID of the server node the RSOE process is for.
- **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.
- **TLS type –** choose RSA.
- **Trusted certificate –** if the certificate file includes multiple certificates, choose whether to trust a single certificate or all of them.

4. Click **Next**.

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 RSOE tab.

RSOE Start Options Reference

Review available RSOE start options. These options affect RSOE logging.

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 synchronization session. • 2 – Request level logging. Provides a more detailed view of HTTP requests within a synchronization session.
Reconnect delay	5	When a connection fails, how long, in seconds, to wait before retrying.

Option	Default	Description
Maximum output file size	10240	Sets the maximum log file output size.
Truncate log file	None	Determines whether to delete the log file contents before adding new messages to the log upon RSOE startup.

Generating the Relay Server Outbound Enabler Configuration File

To quickly and easily replicate a standard configuration to multiple hosts, generate a `rsoeconfig.xml` file.

An administrator might use Sybase Control Center to configure an initial RSOE for development. Once a configuration proves valid and stable, the administrator can generate this file before using `regRelayServer.bat` to apply it to other server nodes .

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

Managing Configured RSOEs

Manage RSOEs you have configured.

Retrieving RSOE Logs

Retrieve RSOE logs from the RSOE host and copy them to another location. Only one log file can be retrieved at a time.

1. In the navigation pane, click **Servers** > **<ServerNode>** > **Server Configuration**.
2. In the administration pane, select the **RSOE** tab.
3. Select an RSOEs for which you want to retrieve data, then click **Retrieve Log**.
4. Click **Next**.
5. Click **Finish** to save the log and choose the target location for the file.

   You can only save a file that contains RSOE log data. Empty log files cannot be retrieved.

Viewing or Editing RSOE Properties

View or edit configuration properties for a selected RSOE.

Relaunching the RSOE Configuration Wizard

Relaunch the RSOE 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 **RSOE** tab.
3. Select the RSOE configurations you want to view or edit, then click **Properties**.

Configuring RSOE General Properties

Configure general properties for RSOE. These properties 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 **RSOE** tab, then click **New**.
3. Configure these properties:
   - **Farm type –** the type of request managed by the relay server: replication based synchronization (RBS) or messaging based synchronization (MBS).
   - **Unwired sever port –** select the port on which RSOE will manage synchronization requests. The default ports are 2480 for RBS, 2481 for secure RBS, and 5001 for MBS.
   - **Relay server host –** type the relay server host RSOE connects to.
   - **Relay server port –** select the relay server port.
   - **Unwired server farm –** select the name of the server farm.
   - **Server node ID –** select the ID of the server node the RSOE process is for.
   - **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.
   - **TLS type –** choose RSA.
   - **Trusted certificate –** if the certificate file includes multiple certificates, choose whether to trust a single certificate or all of them.
4. Click **Next**.

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 RSOE tab.

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 **RSOE** tab.
3. Select the unnecessary RSOE configurations.
4. Stop the selected RSOEs and click **Delete**.
5. Click **OK** on the confirmation message.

Refreshing the RSOE List

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

1. In the navigation pane, click **Servers** > **<ServerNode>** > **Server Configuration**.
2. In the administration pane, select the **RSOE** tab.
3. Click **Refresh**.

Starting and Stopping RSOE

Start and stop the RSOE process as required. However, 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 **RSOE** tab, select the RSOEs, and:
   - Click **Start**.
   - Click **Stop**.

RSOE Tab Data Reference

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

Column Name	Displays
Server Node ID	The server node id specified during the relay server's configuration.
Unwired Server Port	The port number on the server node uses to synchronize data between the enterprise information system and mobile devices. The default synchronization ports are: • Unsecured RBS: 2480 • Secured RBS: 2481 • Secured MBS: 5001

Column Name	Displays
Farm Type	the type of request managed by the relay server: replication based synchronization (RBS) or messaging based synchronization (MBS).
Unwired Server Farm	the server farm for which the relay server manages requests. This property is case-sensitive. The configured value must match the value defined for the RSOE or the connection fails.
Relay Server Host	either the relay server host name or the host name of the load balancer (if one is used in your environment). Device clients use the host name to establish connections. Other relay servers can use the host name to identify peers if a load balancer is not used.
Status	The state of the RSOE process: stopped, running, or error.
Certificate File	The name of the certificate file uploaded to the Unwired Server node certificate store.
Description	Additional details on the status of the RSOE. 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 for this Relay server outbound enabler. • **Unauthorized. –** Check the server node token of RSOE. • **Unrecognized farm or server node ID. –** The Unwired Server farm or server node not configured in the relay server. • **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 invalid. • **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 for it is wrong. • **Bad request. –** Check the URL suffix syntax. Ensure that the URL suffix starts with '\\' or '/'. • **Error writing HTTP headers –** Check the that the trusted certificate is valid and verify the URL suffix syntax. Something may be misformatted.
Log File	The name and location of the RSOE log file.

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:

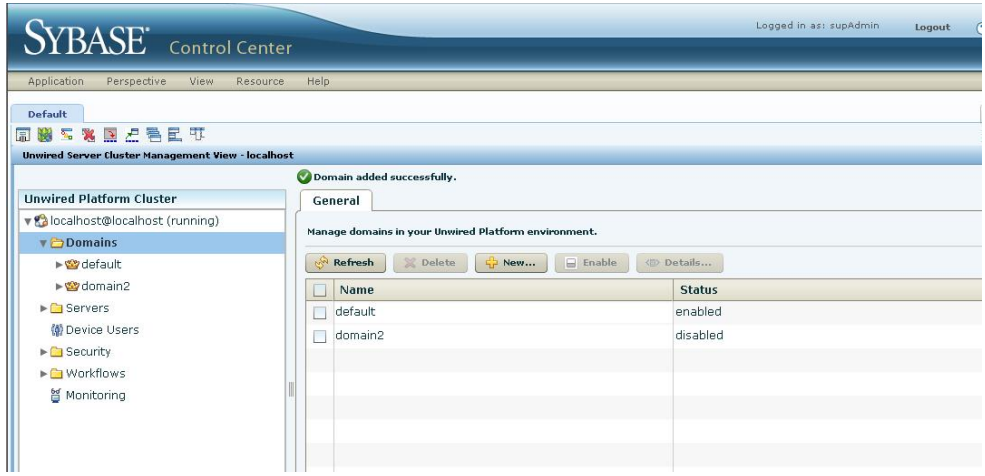
- Create and manage domains using Sybase Control Center from the Unwired Platform administration perspective of Sybase Control Center.
- You can support multiple customers inside the same Unwired Platform cluster.
- You can configure security specifically for individual domains by creating one or more 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.
- Consolidated 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 datasource properties according to their configuration requirements, or create a new monitoring datasource in the "default" domain.

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.



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, expand 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.
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.

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 *Sybase Unwired Platform System Administration Guide > Security Administration > Security Layers > User Security Setup > Security for Administration Users > Setting Up Unwired Platform Roles in Sybase Control Center*.

Task

Perform steps to appropriately configure domain security settings.

Choosing a Security Configuration

Select a security configuration that designates authentication, authorization, attribution, and audit 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, expand the **Domains** folder and select the domain for which you want to choose a named security configuration.
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.

For information on logical and physical roles, see *Logical Roles* and *Physical Roles* in *Sybase Control Center > Configure > Configuring Unwired Platform > Packages > Mapping Roles* .

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.
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.
2. If a role that you require 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. Both 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). Use this option after carefully considering potential consequences.
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.

Domain Administration

Domain administrators interact with Unwired Platform to manage domain artifacts such as packages, subscriptions, connections, and so on.

Sybase Control Center limits access to domain administrator to only those domains that the login has been assigned access to. This requires the platform administrator to:

- Register users' logins as domain administrators.
- Assign the necessary physical role to the domain administrator login in the underlying security provider in the **admin** security configuration.
- Enforce the authorization control by mapping the physical role to the 'SUP Domain Administrator' logical role in the **admin** security configuration in the **default** domain in Sybase Control Center.

By default the supDomainAdmin login can be used to access the **default** domain. Setup requirements vary depending on the environment you administer:

- If you are administering a personal or enterprise development environment or an OpenDS LDAP server, the login is already configured with the SUP Domain Administrator physical role. This phyisical role is then automatically mapped to the SUP Domain Administrator logical role in the **admin** security configuration in the **default** domain.
- If you are administering a deployment environment, you must set up the physical role mapping for the SUP Domain Administrator logical role and ensure that the logins that need domain administration access are assigned membership to one of the mapped physical roles. Once that is configured, only logins with the mapped physical role have access to Unwired Server as domain administrator.

Note: Sybase recommends that you reserve the **default** domain for internal use, and do not share it with tenants.

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.

Domain Logs

Domain logs provide several logical views of domain-level activities involving packages, MBOs, users, devices, operations, and subscriptions. Domain administrators use this data to monitor and troubleshoot activity within an individual domain. However, only a platform administrator can configure domain log properties. The domain logging option must be enabled on the domain or for package-level logging to start.

To ensure domain log properties are selected correctly, the domain and platform administrators must coordinate how to choose the correct domain log configuration for the domain.

Enabling and Configuring Domain Logging

Activate or deactivate domain logging in Sybase Control Center, and configure domain log autopurge settings for all nodes in a cluster. Domain logging collects data that pertains to the activities of all packages in a domain. You must have administrator privileges to configure domain logging.

First, domain-level logging must be enabled by a platform administrator. Domain-level logging controls whether package-level logging captures data. Then either the platform administrator or the domain administrator can enable logging on a per-package basis from the Packages node of Sybase Control Center.

1. In the left navigation pane, expand the **Domains** folder and select the domain for which to configure log settings.
2. Select **Log**.
3. In the right administration pane, select the **Settings** tab.
4. Select one of:
   - **Enable** – activate domain logging in Sybase Control Center.
   - **Disable** – turn off domain logging.
5. Set the autopurge threshold by entering the length of time (in days) to retain domain log data.
6. Click **Save**.

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**.

Packages

Packages are collections of MBOs that are related by application use and authorship and grouped according to maintenance or distribution. Packages are initially created by developers, but are deployed and maintained on a production Unwired Server by administrators.

Administrators cannot change the name of a package if one has been defined by the development team. You can, however, create new package versions when you make an upwardly incompatible change to an existing application. In this case, leave both versions of the package running until every one of the remote client applications has been upgraded to the latest version; only then should you delete the old package version.

Although each mobile business object (MBO) type has unique properties and data sources, MBOs within a package used by an application may be of different types.

Note: You must deploy a package before you can configure or manage it. Package administration tasks vary, depending on the type of package you deploy.

Replication-Based Synchronization Packages

Replication-based synchronization (RBS) packages are packaged mobile business objects (MBOs) that use a replication paradigm to synchronize data and propagate transactions between a device and Unwired Server. RBS package data can be synchronized through a server-initiated push notification transmitted at defined intervals, or based upon the occurrence of a synchronization event.

Administration for RBS packages involves configuring role mapping, setting required properties for cache and synchronization groups, purging the cache, managing push subscription templates and properties, and reviewing client activities.

Messaging-Based Synchronization Packages

Messaging-based synchronization (MBS) packages are packaged mobile business objects (MBOs) that synchronize data through a messaging paradigm. Messaging-based synchronization packages operate through device subscription to the package. Once

subscription is established, the server is responsible for maintaining the subscribed data set through data pushes.

MBS package administration involves configuring role mapping, setting required properties for cache and synchronization groups, purging the cache, managing messaging subscription properties, registering user devices, modifying settings for registered user devices, and reviewing client activities.

Note: Depending on the type of messaging package you are configuring (for example, one that requires a SAP DOE connection, such as a Sybase Mobile Workflow package), not all configuration steps apply because some features are not supported for this package type. For example, Sybase Mobile Workflow packages need not configure cache settings or sync groups. Role mappings are also not required.

DOE-C Packages

Sybase Mobile Workflow for SAP Business Suite and Sybase Mobile Sales for SAP CRM work with Unwired Platform to make parts of SAP Workflow available on your mobile device using SAP Data Orchestration Engine connector (DOE-C) packages.

DOE-C packages implement messaging-based synchronization, which means that they synchronize data through a messaging paradigm. For more information on messaging-based synchronization, see *Sybase Control Center 3.0.2 > Configuring > Configuring Unwired Platform > Packages > Messaging-Based Synchronization Packages*.

Setting the Bulk Load Timeout Property

The Subscribe bulk load timeout property is a package level property targeted to BlackBerry clients for initial server-side subscription operations.

Server-side subscription improves performance and is enabled on the client if the device has a secure digital (SD) memory card enabled. The timeout allows you to set an initial subscription push timeout period. If the timeout period is reached, Unwired Server sends the database file to the device, whether the initial subscribe is complete or not. The timeout window signals that the device has received sufficient import messages to send the server-built database to the client.

Note: This option is only available for Sybase SAP Data Orchestration Engine Connector (DOE-C) packages.

1. In the left navigation pane of Sybase Control Center, expand the Packages folder and select the package to configure.
2. In the right administration pane, select the **Settings** tab.
3. Set the timeout value. The default value is 3600 seconds.

   In addition to the timeout value, you can define the **Subscribe Bulk Load Thread Pool Size** – the maximum number of threads allocated to initial server-side subscription operations. The default value is 5. Setting the thread pool size too high can impact performance. This is a server-side setting that can be set:

a) In the left navigation pane, expand the **Servers** folder and select a server.
b) Select **Server Configuration**.
c) In the right administration pane, select the **General** tab.
d) From the menu bar, select **Performance Configuration**.
e) Expand **Show Optional Properties**.
f) Restart Unwired Server if you change the **Subscribe Bulk Load Thread Pool Size** value for it to take effect.

Enabling and Disabling a Package

Enable or disable a package to allow or prohibit device access to the package. Disabled packages are still available to Sybase Control Center for Unwired Server. By default, all packages are enabled.

When you disable a package, the server unloads all of its elements from memory. Disabling a package prevents the Unwired Server from loading that package at start-up.

Note: You cannot disable a SAP Data Orchestration Engine Connector (DOE-C) package.

1. In the left navigation pane, click **Packages**.
2. In the right administration pane, select the **General** tab.
3. Select the box adjacent to the package you want to enable or disable.

   You can select more than one package to apply the same change to multiple components.
4. Depending on the current status of the package, perform one of:

   - Enable – if the package listed shows a status of disabled, click **Enable**.
   - Disable – if the package listed shows a status of enabled, click **Disable**. The package remains disabled until you or another administrator enables it; restarting Unwired Server does not enable the package.

Enabling Package Logging

Information, errors and events can be recorded for packages.

Prerequisites

Package log data is sent to the domain log, only if you enable domain logging. Ensure you enable domain logging before enabling package logging.

Task

1. In the left navigation pane of Sybase Control Center, expand the **Packages** folder and select the package to configure.
2. In the right administration pane, click the **Settings** tab.
3. Enable package logging or synchronization tracing as required.
4. Click **Save**.

View, search, and export package log data from the domain log.

Enabling Synchronization Tracing

Enable package execution tracing to collect detailed synchronization data in the server log. This mechanism useful for debugging and diagnostics purposes only and has an adverse impact on performance.

Tracing data consists of package synchronization data, such as data upload and download sizes.

1. In the left navigation pane, expand the **Packages** folder and select the package you want to configure.
2. In the right administration pane, click the **Settings** tab.
3. Select **Enable synchronization tracing**.
4. Click **Save**.

   View synchronization tracing data from the server log. See *Sybase Control Center online help > Manage > Managing Unwired Platform > Routine System Maintenance Tasks > Checking Server Logs in Sybase Control Center*.

Selecting a Security Configuration for a Package

Designate a security configuration for a package in Sybase Control Center. This is a required step during package deployment, but you can later change the security configuration.

The administrator must create a security configuration in the cluster and assign it to the domain where the package is deployed before the deployer can assign the security configuration to the package.

1. In the left navigation pane, expand the **Packages** folder, and select the package to configure.
2. In the right administration pane, click the **Settings** tab.
3. Select a security configuration.

   The security profiles that appear in this list have been created by a platform administrator and assigned to the domain.
4. Click **Save**.

Configuring a Cache Group

Select and configure a cache group. The cache is part of the Unwired Server consolidated database (CDB) that is used to store data that is uploaded and downloaded from EIS servers and mobile clients during synchronization.

Cache group configuration differs depending on whether the cache group is defined as "on demand" or "scheduled" during development.

MBO Data in the CDB

A data cache is a copy of MBO data that is stored in a specific area of the consolidated database (CDB). It is used as the data repository for replication and messaging MBOs that are deployed to Unwired Server. "CDB" and "cache" and "MBO data" can sometimes used interchangeably, even though the CDB includes runtime data as well.

When cache data is updated (either with an on-demand or scheduled cache refresh), the remote client database eventually retrieves the updated data from the server's copy of MBO data in the CDB by synchronization.

By giving applications a normalized and uniform view of corporate data, organizations can:

- Lower the barrier to data behind corporate firewalls
- Support development of mobile applications that interact with multiple enterprise back-ends
- Reduce back-end load caused by device client requests

Configuring On Demand Cache Group Properties

Specify the duration of cache data validity by configuring Unwired Server updates to mobile business object (MBO) data for an on demand cache group.

Note: The developer configures a cache group as either on demand or scheduled. If the cache group is "scheduled," the Cache tab is not configurable in the Cache Group Properties dialog.

1. In the left navigation pane, expand the **Packages** folder, and select the package for which you want to configure cache settings.
2. In the right administration pane, click the **Cache Group** tab.
3. Select the cache group you want to configure and click **Properties**.
4. In the **Cache Properties** dialog, enter an expiry for the **Cache Interval** in seconds, minutes, or hours.

   The cache interval determines how frequently Unwired Server updates the consolidated database with changes to enterprise data. See *On Demand Cache Refreshes* in the list of links below.
5. Click **OK**.

On Demand Cache Refreshes

Cache groups designated as "on demand" during development use cache intervals to balance how frequently the object updates enterprise data with the amount of network traffic required to maintain that data.

Unwired Server keeps a local copy of enterprise data in the consolidated database (CDB), and uses an intricate mechanism to manage updates between the CDB and the EIS servers. When data is updated, the remote client database eventually gets updated data from this local copy in the CDB. The caching mechanism allows MBOs to retrieve updated data even if back-end servers fail.

You must choose an appropriate cache interval for your system, since this value determines how frequently the CDB is udpated with data from the EIS. The cache interval must be configured according to business needs. A higher value for the cache may retain stale data, however, a lower value increases the backend EIS load and may impede the client application's performance, because Unwired Server queries the back-end information servers more frequently to look for changes and possibly update the CDB copy.

Frequent queries typically put a higher load on the servers, as well as use more network bandwidth on the server side. While the cache interval does not affect the bandwidth required between the synchronization server and device client applications, nor the performance characteristics of the client applications, the interval you choose can delay synchronization if Unwired Server must first update many records in the CDB.

For example, if the cache interval is 0, each time a client application synchronizes, there is a pause while the Unwired Server rereads data from the EIS and updates the CDB. If, however, the cache interval is greater than 0, then the wait time depends on how long ago the data was refreshed. If the synchronization falls within a recent cache update, synchronization is almost immediate.

Configuring Scheduled Cache Group Properties

Specify the duration of cache data validity by configuring Unwired Server updates to mobile business object (MBO) data for a scheduled cache group.

Prerequisites

You can configure a schedule refresh for a cache only if the developer enables the cache group as "scheduled" during development. Otherwise, the Schedule tab is not configurable in the Cache Group Properties dialog.

Task

1. In the left navigation pane, expand the contents of the **Packages** folder and select the package for which you want to display properties.
2. In the right administration pane, click the **Cache Group** tab.
3. Select the cache group you want to configure and click **Properties**.
4. In the **Schedule** tab of the Cache Properties dialog, set the frequency of the refresh by selecting an appropriate **Schedule Repeat**: hourly, daily or custom.

   This property determines what other schedule properties you must configure. Each option is documented in a seperate topic which further discusses the details for each frequency type. For more details, see the corresponding topic.

Scheduling an Hourly or Daily Refresh

Scheduling an hourly or daily cache refresh means that information is fetched from the enterprise information server (EIS) and populated into the cache on either of these hourly or daily frequencies according to the schedule and range of time you configure.

The Schedule tab in the Cache Property dialog displays options appropriate for configuring this type of schedule.

1. Select either **Hourly** or **Daily** as the **Schedule Repeat** criteria.
2. (Optional) If you want to set a range to control which days the schedule refresh runs, configure a start date and time, an end date and time, or day of week (if applicable).

   - Select **Start Date** to set a date for which the first execution of the scheduled refresh is performed. To be more specific, you can also select **Start Time** to specify a start time. In this case, the refresh cannot begin until a given time on a given day has been reached. A start date and time are inclusive.

     If you do not set a start date and time, then, by default, the date and time that Unwired Server starts is used.
   - Select **End Date** to set a date that ends the repeating refresh transactions for a package. To be more specific, you can also select **End Time** to specify an end time. An end date and time are exclusive. This means that a refresh transaction runs up to, but does not include, the end time. For example, if a schedule has a start time of 13:00 and an end time of 16:00 and repeats every hour, it runs at 13:00, 14:00, and 15:00, but not at 16:00.

     If you do not set an end date and time, then, by default, the date and time that Unwired Server stops is used.
   - Select **Specify Week Days** to select the days of the week that the refresh transaction runs. This means that for the days you select, the refresh runs every week on the day or days you specify. A weekday is inclusive. This means that any day you choose is included in the frequency. All others are excluded.

   When the schedule expires, the automatic refresh you configured terminates, unless the end user initiates a refresh.
3. Click **Save**.

Scheduling a Custom Refresh

Scheduling a custom cache refresh is the most flexible of all cache refresh schedules. This means that information is fetched from the enterprise information system (EIS) according to the schedule repeat interval you specify.

The Schedule tab in the Cache Property window displays options appropriate for configuring this type of schedule.

1. Select **Custom** as the schedule repeat criteria.

2. Specify a repeat **Interval**, in minutes or seconds, to determine how often the cache refresh occurs.

   This interval determines how frequently Unwired Server updates the consolidated database with changes to enterprise data. The default is 0 seconds, which means the mobile business object retrieves the data from the enterprise information server (EIS) on every playback request. If you choose something other than 0 seconds, the data is held by the cache for the duration of the specified interval.
3. (Optional) To set a range to control which days the schedule refresh runs, configure a start date and time, end date and time, or day of week (if applicable).

   - Select **Start Date** to set a date for which the first execution of the scheduled refresh is performed. To be more specific, you can also select **Start Time** to specify a start time. In this case, the refresh cannot begin until a given time on a given day has been reached. A start date and time are inclusive.

     If you do not set a start date and time, then, by default, the date and time that Unwired Server starts is used.
   - Select **End Date** to set a date that ends the repeating refresh transactions for a package. To be more specific, you can also select **End Time** to specify an end time. An end date and time are exclusive. This means that a refresh transaction runs up to, but does not include, the end time. For example, if a schedule has a start time of 13:00 and an end time of 16:00 and repeats every hour, it runs at 13:00, 14:00, and 15:00, but not at 16:00.

     If you do not set an end date and time, then, by default, the date and time that Unwired Server stops is used.
   - Select **Specify Week Days** to select the days of the week that the refresh transaction runs. This means that for the days you select, the refresh runs every week on the day or days you specify. A weekday is inclusive. This means that any day you choose is included in the frequency. All others are excluded.

   When the schedule expires, the automatic refresh you configured terminates, unless the end user initiates a refresh.
4. Click **Save**.

Scheduled Cache Refreshes

A schedule-driven cache refresh is a background task that runs between a configured start and endpoint at scheduled intervals during normal server operation.

A schedule-driven cache refresh defines a contract between Unwired Server and back-end information servers. Normally, data is retrieved from a server (for example, a database, and an SAP repository, or a Web service) when a device user synchronizes. If the administrator wants the data to be preloaded, he or she configures the Unwired Server repeat interval to expedite data updates on the device.

Two properties configure the cache refresh schedule, which is used with a subscription to synchronize data for mobile business objects (MBOs).

- Schedule repeat – determines the time frame when data is refreshed. If you set up a schedule to repeatedly refresh data, information is always refreshed. Set the schedule to meet business application requirements for data consistency.

  As an administrator, you may also use a schedule repeat to look for data changes and alert subscribed clients to synchronize when there are changes. Keep in mind, however, that the actual detection of changes and sending of data (MBS) or notifications (RBS) depends on the "change detection interval" property of the synchronization groups in the package, as well as the notification threshold property of subscriptions of RBS package or push related device settings for MBS package.
- Repeat interval – determines how often Unwired Server updates the cache with changes to backend data.

Online Refresh Policy

MBOs that use an Online refresh policy indicates that the MBOs are to be used only in Workflow applications where access to real-time enterprise information system (EIS) data is required (cache validity is zero).

Data is valid in the Unwired Server cache only until delivery and immediately invalid. You cannot modify the cache or schedule of the Online policy. Expired data is purged from the cache based on a schedule at the domain level.

DCN Refresh Policy

The cache refresh and schedule options are disabled for DCN (data change notification) policy, since data never expires and is not refreshed based on client demand or a schedule.

Cache data does not expire until a cache invalidate operation is invoked or a data change notification request is received from the enterprise information system (EIS).

Purging a Cache Group

Physically delete data that has been logically deleted from the cache. Cached data is marked as logically deleted when certain activities occur in the client application or back end.

1. In the left navigation pane of Sybase Control Center, expand the **Packages** folder and select the package to configure.
2. In the right administration pane, select the **Cache Group** tab.
3. Click **OK**.

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.

For information on logical and physical roles, see *Logical Roles* and *Physical Roles* in *Sybase Control Center > Configure > Configuring Unwired Platform > Packages > Mapping Roles* .

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.
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.
2. If a role that you require 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. Both 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). Use this option after carefully considering potential consequences.
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.

Logical Roles

Default logical roles are administration roles already setup in Unwired Platform. Other roles will typically be defined by package developers, and allow developers to define identities that each application uses to indicate access rights to its different objects. Logical roles may or may not replicate physical names already defined for your security provider.

Logical roles allow secured access to Unwired Platform resources for several users at once, and you can define them for one or more packages in a domain. If a developer has created a logical role for a package, you may need to manually map it to one or more physical roles.

In the absence of explicit mapping, the default role mapping is set to AUTO, which is equivalent of logical role mapping to a physical role of the same name, in the underlying provider of that security configuration.

The following default roles are used: SUP Administrator, SUP Domain Administrator, and SUP DCN User. These roles must be mapped. In a development environment, mappings are automatically created. In a production environment, physical roles must be added manually to the directory used, and then manually mapped in Sybase Control Center.

SUP DCN User Role

The SUP DCN User is a logical role that Unwired Platform uses to authorize any DCN event: updating data in the cache, executing an operation, or triggering a workflow package.

Before any DCN event is submitted, the person or group mapped to this role must be authenticated and authorized by the security configuration used. By default, SUP DCN User is automatically available to all new security configurations you create. However, the underlying default varies depending on the environment in use.

- In a development environment – A physical role called DCNRole is automatically added to the OpenDS LDAP directory. The SUP DCN User logical role is mapped to the DCNRole

physical role, which is group automatically added to this directory. Both the supAdmin and supDcnDeveloper LDAP users are added as members of DCNRole, so either user may perform DCN.

- In a production environment – In a single domain environment, the "default" Domain's security configuration has "no security" set by default. That means, any user/password credentials are authenticated, and all roles are granted to everyone. So any user could perform DCN initially.

  However, eventually this default configuration for the default domain will change. In this case—as in the case for any additional security configurations that is added—the SUP DCN User logical role must be mapped to some physical role in the backend security systems, and the user who performs DCN must be in that physical role.

  Note: If security configuration provider does not support roles, you must perform a special mapping manually. You can create a single user role (a role explicitly mapped to a single user authenticated against the particular security configuration). This is achieved by prefixing the username with `user:`. For example, a mapped physical role named `user:joep` would authorize the user named 'joep' to issue DCN to any package associated with the particular security configuration that contains this mapping, or to issue a workflow DCN to any user authenticated against the particular security configuration.

  To map the SUP DCN User to a user in the underlying security repository, the user name must be first defined in Sybase Control Center as a physical role that is mapable. Then, SUP DCN User role can be mapped to a physical user or to a physical role from Sybase Control Center. For example, if you want to map SUP DCN User to a user use the format `user:<User>` . Alternatively, you can also map it to a role with `<PhysicalRole>`.

  If you are supporting multiple domains, then the user name also needs to include the named security configuration for the package the DCN is targeted for, by appending *@<DomainSecurityConfigName>* as a suffix to that name. Suppose you have two packages (PKG_A, PKG_B) deployed to 2 domains (Domain_A, Domain_B) respectively. Further, assume that PKG_A in Domain_A has been assigned to the "admin" security configuration, whereas PKG_B in Domain_B has been assigned to the "alternateSecurityConfig" security configuration.

  - A user doing DCN to PKG_A should identify themselves as *User@admin*.
  - A user doing DCN to PKG_B should identify themselves as *User@alternateSecurityConfig*.

  If you are using ActiveDirectory, and are using email addresses for user names, then definitions appear as *<username@myaddress>@<DomainSecurityConfigName>*.

Furthermore, the implementation varies depending on the DCN service used:

- For workflows, because the resource the user is pushing data towards is a group of named users (users authenticated previously successfully against a certain security configuration), therefore the user must have the authorization to push to that particular security configuration. The user must have be mapped to SUP DCN User in the security configuration for the workflow target.

- A user having SUP DCN User logical role in security configuration 'mySecConfig1' must not have the right to push workflow DCN or regular DCN to a user or package associated with 'mySecConfig2'.

Physical Roles

Physical roles are named references to roles or groups that an administrator has defined on a back-end enterprise security provider. Mapping a logical role to a physical role allows authorization control in the Unwired Server. Replicate these names exactly, so that logical roles can be mapped correctly to the server role.

Configuring a Synchronization Group for RBS Packages

Determine the frequency with which push notifications are generated by Unwired Server for synchronization of mobile business objects (MBOs). The delivery time is determined by this equation: `delivery of push notification = schedule repeat value + synchronization group change detection interval value + notification threshold value`. There may be an additional 10 second delay, because the internal thread for a cache refresh from DCN or cache group that performs the change detection runs every 10 seconds.

A synchronization group is a collection of MBOs that are synchronized together. For replication-based synchronization packages (RBS), the synchronization group defines the logical unit of synchronization and data notifications. The RBS client receives notifications when data change is detected for any of the MBOs in a synchronization group, which subsequently results in the client synchronization frequency of those MBOs.

1. In the left navigation pane, expand the **Packages** folder and select the package containing the sync group you want to configure.
2. Select the desired sync group and click **Properties**.
3. Select a change detection interval. This value determines how frequently Unwired Server looks for changes to MBOs, and generates push notifications. The default is 1 hour.
4. Click **OK**.

Configuring a Synchronization Group for MBS Packages

Determine the frequency with data changes are generated by Unwired Server for synchronization of mobile business objects (MBOs). The delivery time is determined by this equation: `delivery of data change = schedule repeat value + change detection interval value + + device push settings`. There may be an additional 10 second delay, because the internal thread for performing the change detection runs every 10 seconds.

A synchronization group is a collection of MBOs that are synchronized together. For messaging-based synchronization (MBS) packages, the unit of changed data for MBOs belonging to the synchronization group is sent directly to clients.

1. In the left navigation pane, expand the **Packages** folder and select the package containing the sync group you want to configure.
2. Select the desired sync group and click **Properties**.
3. Select a change detection interval. This value determines how frequently Unwired Server looks for changes to MBOs, and sends messages. The default is 1 hour.
4. Click **OK**.

Configuring Replication Package Subscriptions

Configure subscriptions and subscription templates to allow the device user to be notified when information is available, depending on the subscription properties configured for a package. Subscription templates allow you to configure predefined properties for a synchronization group. A client's first synchronization for a specified synchronization group results in subscription creation using both the template and client-specified properties.

There are two main activities to set up a notification for a replication-based synchronization package.

Creating Subscription Templates

Create a subscription template to specify synchronization targets and behavior for subscribed users. A template is useful to create a set of predefined values that are used frequently. Otherwise, a subscription is still automatically created for each client upon explicit indication of interest for a device notification.

This is an optional step; it is only required if an administrator wants to establish preset subscription properties. The subscription properties can be modified from device application, but only if the **Admin lock** property is disabled.

1. In the left navigation pane, expand the **Packages** folder and select the replication based sync package you want to configure.
2. In the right administration pane, click the **Subscriptions** tab.
3. From the menu bar, select **Templates**.
4. Click **New**.
5. In the New Template dialog, select settings for these options:
   - Synchronization Group – the group of MBOs that a client receives data change notifications for when data changes occur.
   - Notification Threshold – the length of time that must pass since a client's last synchronization before another notification is sent.
   - Admin Lock – (enable or disable) prevents device users from modifying the push synchronization state or sync interval value configured in the subscription. If the admin lock is disabled, the device client user can change these properties, and these changes take effect the next time the client user synchronizes the package to which the subscription applies.

- Push – (enable or disable) if enabled, automatic server-initiated notifications are pushed to users when changes occur in the cache. If disabled, device users perform client-initiated synchronizations when they receive an outbound notification.

Note: If you intend to use push synchronization with BlackBerry devices, enable push synchronization in the BlackBerry server. See the BlackBerry server documentation for details.

6. Click **OK**.
   The new subscription template appears in the list of templates.

Notifications are delivered to BlackBerry device clients using push-based notification settings and to Windows Mobile device clients using pull-based notification settings. The "poll every" setting determines the notification delivery behavior. See *Sybase Control Center Online Help > Configure > Configuring Unwired Platform > Unwired Server > Server Properties > Replication > Configuring Push and Pull Notifications Using HTTP or Lightweight Polling*.

Configuring Subscription Properties

View and configure subscription properties for device users subscribed to deployed replication-based synchronization packages.

1. In the left navigation pane, expand the **Packages** folder and select the replication-based synchronization package you want to configure.
2. In the right navigation pane, click the **Subscriptions** tab.
3. From the menu bar, select **Devices**.
4. Check the box adjacent to a device user and click **Properties** to view these subscription properties:
   - Device ID – the unique identifier attached to the device hardware of a registered device.
   - Activation User Name – the name of the user associated with the device ID.
   - Package Name – the name of the package to which the subscription belongs.
   - Sync Counts – the total number of synchronizations for the subscription since the synchronization history was last cleared.
   - Notification Threshold – the length of time that must pass since a client's last synchronization before another notification is sent.
   - Last Sync Time – the date and time that the last synchronization for the subscription occurred.
   - Synchronization Group – the group of MBOs that a client receives data change notifications for when data changes occur.
   - Admin Lock – (enable or disable) prevents device users from modifying the push synchronization state or sync interval value configured in the subscription. If the admin lock is disabled, the device client user can change these properties, and these changes take effect the next time the client user synchronizes the package to which the subscription applies.

- Push – (enable or disable) if enabled, automatic server-initiated notifications are pushed to users when changes occur in the cache. If disabled, device users perform client-initiated synchronizations when they receive an outbound notification.

5. Check the box adjacent to **Clear sync history** in order to erase stored synchronization details for the subscription.
6. If you made changes to subsciption properties, click **Save**. Otherwise, click **Cancel** to return to the Subscriptions tab.

Configuring Messaging and DOE-C Package Device Settings

View and edit device properties for messaging and SAP Data Orchestration Engine connector (DOE-C) subscriptions that allow you to manage synchronization messages between Unwired Server and mobile device users.

1. In the left navigation pane, expand the **Packages** folder, and select the package you want to configure.
2. In the right navigation pane, click the **Subscriptions** tab.
3. Check the box adjacent to a device user and click **Device Settings** to view these subscription properties:
   - Activation User Name – the name of the user associated with the device ID.
   - Device Type – the type of the messaging device (for example, BlackBerry).
   - Device Model – the manufacturer of the registered mobile device.
   - Device ID – the unique identifier attached to the device hardware of a registered device.
   - Status – the current status of a device. The possible values are: Online, Offline, and Pending Activation.
   - Pending Items – the pending items on the server side that needs to be sent to the device.
   - Last Delivery – the date and time of last item delivered from Unwired Server to the device.
   - Client Version – the Sybase Unwired Platform client runtime version of the client.
   - Node – the node of the cluster on which the device is registered.
4. Click **Next**.
5. Configure these property categories, as required:
   - Connection
   - Custom Settings
   - Device Advanced
   - Device Info
   - Features
   - Scheduled Sync
   - User Registration
   - Apple Push Notifications (iPhone only)

Viewing and Changing Messaging and DOE-C Package Connection Properties

View and edit connection properties for messaging and SAP Data Orchestration Engine connector (DOE-C) packages.

1. In the left navigation pane, expand the **Packages** folder, and select the package whose connection properties you want to view or change.
2. In the right navigation pane, click the **Connection** tab.
3. If you wish to change any of the settings:
   1. In the left navigation pane, click the **Connections** icon.
   2. Select the checkbox for the **Connection Pool Name** that matches the name of the package whose connection properties you were just viewing.
   3. Click **Properties**.
   4. Make desired changes and click **Save**.
   5. In the left navigation pane, click the **Connections** icon.
   6. Select the checkbox for the **Connection Pool Name** that matches the name of the package whose connection properties you wish to test.
   7. Click **Properties**.
   8. Click **Test Connection**.

      If the connection test is not successful, see *Troubleshooting > Troubleshoot the System > Troubleshoot EIS Connections > Connection Test Errors*

Managing Deployed Package Subscriptions

Manage replication, messaging, and SAP Data Orchestration Engine connector (DOE-C) package subscriptions that specify the synchronization messages mobile device users receive.

Subscription management tasks include pinging, unsubscribing, recovering, suspending, resuming, resynchronizing, and logging subscriptions. Subscription tasks vary by the package type.

These subscription management tasks apply only to the package types specified in the table below. Perform each task in the Subscriptions tab of the deployed package you are managing.

Table 8. Subscription management tasks

Subscription task	Description	Summary	Package type
Ping	Ensure that push information a user provides for a device is configured correctly. If the ping is successful, notifications and subsequent data synchronizations occur as defined by each subscription. If the ping fails, open the log and check for an incorrect host name or port number.	Select the box adjacent to the device ID, and click **Ping**.	Replication
Unsubscribe	Remove a subscription from Unwired Server.	Select the box adjacent to the device ID, and click **Unsubscribe** for replication packages, messaging packages, and DOE-C packages. For Windows Mobile, the device application must include the DatabaseClass.CleanAllData(); method for data to be unsubscribed correctly. If this method is not used, **Unsubscribe** and **Subscribe** could work unpredictably.	All
Recover	Reestablish a relationship between the device and Unwired Server. Perform recovery under severe circumstances when a device is unable to successfully synchronize data. During subscription recovery, Unwired Server purges all enterprise data on the device. It retains the device ID and subscription information so that all data can then be resynchronized and loaded onto the device.	Check the box adjacent to the subscription ID of the device, and click **Recover**.	Messaging

Subscription task	Description	Summary	Package type
Suspend/resume	Control the deactivation and reactivation of package subscriptions: • Suspend – temporarily block data synchronization for a device subscribed to a particular package. • Resume – reactivate a package subscription after it has been suspended.	Select the box adjacent to the subscription ID of the device, and click either **Suspend** or **Resume**.	Messaging DOE-C
Resynchronize	Reactivate subscriptions to a deployed package. If a DOE-C subscription does not respond to the SAP DOE quickly enough, the DOE may mark that subscription's queues as "blocked" and stop sending messages to the DOE-C. Resynchronize to resume communication from the DOE to the DOE-C subscription.	Check the box adjacent to the subscription ID of the device, and click **ReSync**.	DOE-C
Purge	Removes subscriptions that are no longer referenced by any active users.	Select the subscription, click **Purge**, and then select the criteria.	Messaging Replication

Reviewing Replication Package Subscriptions

Review information on replication-based synchronization package subscriptions that allows you to manage the notifications that are sent to device users, depending on the synchronization group to which they belong.

In the left navigation pane, select a deployed package. In the administration console, click the **Subscriptions** tab to review these properties:

- Device ID – the unique identifier attached to the device hardware of a registered device.
- Activation User Name – the name of the user associated with the device ID.
- Synchronization Group – the group of MBOs that a client receives data change notifications for when data changes occur.

Reviewing Messaging Package Subscriptions

Review information on messaging-based synchronization package subscriptions in order to manage the synchronization data that device users receive.

In the left navigation pane, select a deployed package. In the administration console, click the **Subscriptions** tab to review these properties:

- Device ID – the unique identifier attached to the device hardware of a registered device.
- Activation User Name – the name of the user associated with the device ID.
- Status – the current status of a device. The possible values are: Running, Suspended, Pending Activation, Online, Offline, and Expired.
- Last Server Response Time – the date and time that the last outbound response was sent from Unwired Server to the client.
- Client ID – the device application ID, which identifies the package database for the application.
- Application Name – the name of the device application used by the subscription.

Reviewing DOE-C Package Subscriptions

Review information on SAP Data Orchestration Engine connector (DOE-C) package subscriptions in order to manage the synchronization data that device users receive.

In the left navigation pane, select a deployed package. In the administration console, click the **Subscriptions** tab to review these properties:

- Device ID – the unique identifier attached to the device hardware of a registered device.
- Activation User Name – the name of the user associated with the device ID.
- Last Server Response Time – the date and time that the last outbound response was sent from Unwired Server to the client.
- Client ID – the device application ID, which identifies the package database for the application.
- Application Name – the name of the device application used by the subscription.
- Status – the current status of a device. The possible values are: Running, Suspended, Pending Activation, Online, Offline, and Expired.
- Packet Dropped – the current packet dropped state of a device. The values are true or false.

Select Advanced to view these properties:

- Subscription ID – the unique identifier of the subscription.
- Logical ID – the unique identifier of a registered device that is generated and maintained by Unwired Server.

Connections

Connections allow Unwired Server to communicate with data sources. To facilitate the connection process, define a set of properties for each data source.

A connection is required to send queries to mobile business objects, and to receive answers. The format in which data is communicated depends on the type of data source; for example, database data sources use a result set, while Web services data sources provide XML files, and SAP data sources use tables.

Establish connections by supplying an underlying driver and a connection string. Together, the driver and string allow you to address the data source, and provide you a mechanism by which to set the appropriate user authentication credentials and connection properties that describe the connection instance. Once a connection is established, Unwired Server can open and closed it as required.

Connection pools

Unwired Server maintains database connections in a connection pool, which is a cache of database connections for the consolidated database (CDB) 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 not 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.

Creating Connections and Connection Templates

Create a new connection or connection template that defines the properties needed to connect to a new data source.

1. In the left navigation pane, expand the **Domains** folder, and select the domain for which you want to create a new connection.
2. Select **Connections**.
3. In the right administration pane:
   - To create a new connection – select the **Connections** tab, and click **New**.
   - To create a new connection template – select the **Templates** tab, and click **New**.
4. Enter a unique **Connection pool name** or template name.
5. Select the **Connection pool type** or template type:
   - JDBC – choose this for most database connections.
   - WS – choose this if you are connecting to a Web Services (SOAP or REST) data source.
   - SAP – choose this if you are connecting to an SAP (JCO) datasource.

- DOEC – choose this if you are connecting to an SAP (DOE) datasource

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 sambledb 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.

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.

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 *Troubleshooting > Troubleshoot the System > Troubleshoot EIS Connections > Connection Test Errors*.

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) .

JDBC Properties

Configure Java Database Connectivity (JDBC) connection properties.

This list of properties can be used by all datasource types. Sybase does not document native properties used only by a single driver. However, you can also use native driver properties, naming them using this syntax:

```
<driver_type>:<NativeConnPropName>=<SupportedValue>
```

Note: If Unwired Server is connecting to a database with a JDBC driver, ensure you have copied required JAR files to correct locations. See *Preparing Unwired Server to Connect to JDBC Databases* in the *System Administration* guide.

Name	Description	Supported values
After Insert	Changes the value to `into` if a database requires `insert into` rather than the abbreviated `into`.	`into`
Batch Delimiter	Sets a delimiter, for example, a semicolon, that can be used to separate multiple SQL statements within a statement batch.	`<delimiter>`
Blob Updater	Specifies the name of a class that can be used to update database BLOB (long binary) objects when the BLOB size is greater than psMaximumBlobLength.	`<class name>` The class must implement the `com.sybase.djc.sql.BlobUpdater` interface.
Clob Updater	Specifies the name of a class that can be used to update database CLOB (long string) objects when the CLOB size is greater than psMaximumClobLength.	`<class name>` The class must implement the `com.sybase.djc.sql.ClobUpdater` interface.

Name	Description	Supported values
Code Set	Specifies how to represent a repertoire of characters by setting the value of CS_SYB_CHARSET for this datasource. Used when the data in the datasource is localized. If you do not specify the correct code set, characters may be rendered incorrectly.	`[server]` If the value is server, the value of the current application server's defaultCodeSet property is used.
Commit Protocol	Specifies how Unwired Server handles connections for a datasource at commit time, specifically when a single transaction requires data from multiple endpoints. If you use XA, the recovery log is stored in the tx_manager datasource, and its commit protocol must be optimistic. If tx_manager is aliased to another datasource (that is, one that is defined with the aliasFor property), the commit protocol for that datasource must be optimistic. A last-resource optimization ensures full conformance with the XA specification. The commit protocol for all other datasources should be XA_2PC. Alternately, a transaction that accesses multiple datasources for which the commit protocols are optimistic is permitted.	`[optimistic| pessimistic | XA_2PC]` Choose only one of these protocols: • Optimistic – enables connections to be committed without regard for other connections enlisted in the transaction, assuming that the transaction is not marked for rollback and will successfully commit on all resources. Note: if a transaction accesses multiple data sources with commit protocol of "optimistic", atomicity is not guaranteed. • Pessimistic – specifies that you do not expect any multi-resource transactions. An exception will be thrown (and transaction rolled back) if any attempt is made to use more than one "pessimistic" data source in the same transaction. • XA_2PC – specifies use of the XA two phase commit protocol. If you are using two phase commit, then the recovery log is stored in the "tx_manager" data source, and that data source (or the one it is aliased to) must have the commit protocol of "optimistic" or "pessimistic". All other data sources for which atomicity must be ensured should have the "XA_2PC" commit protocol.

Name	Description	Supported values
Datasource Class	Sets the class that implements the JDBC datasource. Use this property (along with the driverClass property) only if you do not have a predefined database-type entry in Unwired Server for the kind of SQL database you are connecting to. For example, you must use this property for MySQL database connections. You can implement a datasource class to work with a distributed transaction environment. Because Unwired Server supports distributed transactions, some datasources may require that a datasource class be implemented for Unwired Server to interact with it. For two-phase transactions, use the xaDataSourceClass connection property instead.	`<com.`*`mydata-source`*`.jdbc.Driver>`
Database Command Echo	Echoes a database command to both the console window and the server log file. Use this property to immediately see and record the status or outcome of database commands. When you enable this property, Unwired Server echoes every SQL query to `ml.log`, which may help you debug your application.	`[true|false]` Set a value of 1 to echo the database commands like `databaseStartCommand`, and `databaseStopCommand`. Otherwise, do not set this property, or use a value of 0 to disable the echo.

Name	Description	Supported values
Database Create Command	Specifies the operating system command used to create the database for this datasource. If this command is defined and the file referenced by ${databaseFile} does not exist, the command is run to create the database when an application component attempts to obtain the first connection from the connection pool for this datasource.	`<command>` Example: `<UnwiredPlatform_InstallDir>\Servers\SQLAnywhere11\BIN32\dbinit -q ${databaseFile}`
Database File	Indicates the database file to load when connecting to a datasource. Use this property when the path to the database file differs from the one normally used by the database server. If the database you want to connect to is already running, use the databaseName connection parameter.	`<string>` Supply a complete path and file name. The database file you specify must be on the same host as the server.

Name	Description	Supported values
Database Name	Identifies a loaded database with which to establish a connection, when connecting to a datasource. Set a database name, so you can refer to the database by name in other property definitions for a datasource. If the database to connect to is not already running, use the databaseFile connection parameter so the database can be started. **Note:** For Unwired Server, you typically do not need to use this property. Usually, when you start a database on a server, the database is assigned a name. The mechanism by which this occurs varies. An administrator can use the DBN option to set a unique name, or the server may use the base of the file name with the extension and path removed.	`[DBN|default]` If you set this property to default, the name is obtained from the DBN option set by the database administrator. If no value is used, the database name is inherited from the database type.
Database Start Command	Specifies the operating system command used to start the database for this datasource. If this command is defined and the database is not running, the command is run to start the database when the datasource is activated.	`<command>` Example: `<UnwiredPlatform_InstallDir>\Servers\SQLAnywhere11\BIN32\dbsrv11.exe`

Database Stop Command	Specifies the operating system command used to stop the database for this datasource. If this property is defined and the database is running, this command executes during shutdown.	`<command>`
		For a Adaptive Server™ Anywhere database, where the user name and password are the defaults (dba and sql), enter:
		`<UnwiredPlatform_InstallDir>\Servers\SQLAnywhere11\BIN32\dbsrv11.exe`

Name	Description	Supported values
Database Type	Specifies the database type.	`<database type>`
Database URL	Sets the JDBC URL for connecting to the database if the datasource requires an Internet connection. Typically, the server attempts to construct the database URL from the various connection properties you specify (for example, portNumber, databaseName). However, because some drivers require a special or unique URL syntax, this property allows you to override the server defaults and instead provide explicit values for this URL.	`<JDBCurl>` The database URL is JDBC driver vendor-specific. For details, refer to the driver vendor's JDBC documentation.
Driver Class	Sets the name of the class that implements the JDBC driver. Use this property (along with the dataSourceClass property) only if you do not have a predefined database-type entry in Unwired Server for the kind of SQL database you are connecting to. For example, MySQL database connections require you to use this connection property. To create a connection to a database system, you must use the compatible JDBC driver classes. Sybase does not provide these classes; you must obtain them from the database manufacturer.	*`<Class.forName("foo.bar.Driver")>`* Replace <Class.forName("foo.bar.Driver")> with the name of your driver.
Driver Debug	Enables debugging for the driver.	`[true|false]` Set to true to enable debugging, or false to disable.
Driver Debug Settings	Configures debug settings for the driver debugger.	`[default|<setting>]` The default is STATIC:ALL.

Name	Description	Supported values
Initial Pool Size	Sets the initial number of connections in the pool for a datasource. In general, holding a connection causes a less dramatic performance impact than creating a new connection. Keep your pool size large enough for the number of concurrent requests you have; ideally, your connection pool size should ensure that you never run out of available connections. The initialPoolSize value is applied to the next time you start Unwired Server.	`<int>` Replace <int> with an integer to preallocate and open the specified number of connections at start-up. The default is 0. Sybase suggests that you start with 0, and create additional connections as necessary. The value you choose allows you to create additional connections before client synchronization requires the server to create them.
Is Download Zipped	Specifies whether the driver file downloaded from jdbcDriverDownloadURL is in `.ZIP` format. This property is ignored if the value of jdbcDriverDownloadURL connection is an empty string.	`[True|False]` The default is false. The file is copied, but not zipped to `<UnwiredPlatform-install>\lib\jdbc`. Set isDownloadZipped to true to save the file to `<UnwiredPlatform-install>\lib\jdbc` and unzip the archived copy.
JDBC Driver Download URL	Specifies the URL from which you can download a database driver. Use this property with isDownloadZipped to put the driver in an archive file before the download starts.	`<URL>` Replace <URL> with the URL from which the driver can be downloaded.

Name	Description	Supported values
Language	For those interfaces that support localization, this property specifies the language to use when connecting to your target database. When you specify a value for this property, Unwired Server: • Allocates a CS_LOCALE structure for this connection • Sets the CS_SYB_LANG value to the language you specify • Sets the Microsoft SQL Server CS_LOC_PROP connection property with the new locale information Unwired Server can access Unicode data in an Adaptive Server® 12.5 or later, or in Unicode columns in Adaptive Server 12.5 or later. Unwired Server automatically converts between double-byte character set (DBCS) data and Unicode, provided that the Language and CodeSet parameters are set with DBCS values.	`<language>` Replace <language> with the language being used.
Max Idle Time	Specifies the number of seconds an idle connection remains in the pool before it is dropped.	`<int>` If the value is 0, idle connections remain in the pool until the server shuts down. The default is 60.

Name	Description	Supported values
Max Pool Size	Sets the maximum number of connections allocated to the pool for this datasource. Increase the maxPoolSize property value when you have a large user base. To determine whether a value is high enough, look for ResourceMonitorTimeoutException exceptions in `<hostname>-server.log`. Continue increasing the value, until this exception no longer occurs. To further reduce the likelihood of deadlocks, configure a higher value for maxWaitTime. To control the range of the pool size, use this property with minPoolSize.	`<int>` A value of 0 sets no limit to the maximum connection pool size.
Max Wait Time	Sets the maximum number of seconds to wait for a connection before the request is cancelled.	`<int>` The default is 60.
Max Statements	Specifies the maximum number of JDBC prepared statements that can be cached for each connection by the JDBC driver. The value of this property is specific to each JDBC driver.	`<int>` A value of 0 (default) sets no limit to the maximum statements.
Min Pool Size	Sets the minimum number of connections allocated to the pool for this datasource.	`<int>` A value of 0 (default) sets no limit to the minimum connection pool size.

Name	Description	Supported values
Network Protocol	Sets the protocol used for network communication with the datasource. Use this property (along with the driverClass, and dataSourceClass properties) only if you do not have a predefined database-type entry in Unwired Server for the kind of SQL database you are connecting to. For example, you may be required to use this property for MySQL database connections.	The network protocol is JDBC driver vendor-specific. There are no predefined values. See the driver vendor's JDBC documentation.
Password	Specifies the password for connecting to the database.	`[default|<password>]`
Ping and Set Session Auth	Runs the ping and session-authorization commands in a single command batch; may improve performance. You can only enable the Ping and Set Session Auth property if you have enabled the Set Session Auth property so database work runs under the effective user ID of the client.	`[True|False]` Set to true to enable, or false to disable.
Ping Connections	Pings connections before attempting to reuse them from the connection pool.	`[True|False]` Set to true to enable ping connections, or false to disable.
Ping SQL	Specify the SQL statement to use when testing the database connection with ping.	`[default|<statement>]` Replace <statement> with the SQL statement identifier. The default is "select 1".
Port Number	Sets the server port number where the database server listens for connection requests.	`[default|<port>]` Replace <port> with the TCP/IP port number to use (that is, 1 – 65535). If you set the value as default, the default protocol of the datasource is used.

Name	Description	Supported values
PS Maximum Blob Length	Indicates the maximum number of bytes allowed when updating a BLOB datatype using PreparedStatement.setBytes.	`[default|<int>]` Replace <int> with the number of bytes allowed during an update. The default is 16384.
PS Maximum Clob Length	Indicates the maximum number of characters allowed when updating a CLOB datatype using PreparedStatement.setString.	`[default|<int>]` Replace <int> with the number of bytes allowed during an update. The default is 16384.
Role Name	Sets the database role that the user must have to log in to the database.	`[default|<name>]` If you set this value to default, the default database role name of the datasource is used.
Server Name	Defines the host where the database server is running.	`<name>` Replace <name> with an appropriate name for the server.
Service Name	Defines the service name for the datasource. For SQL Anywhere servers, use this property to specify the database you are attaching to.	`<name>` Replace <name> with an appropriate name for the service.
Set Session Auth	Establishes an effective database identity that matches the current mobile application user. If you use this property, you must also use setSessionAuthSystemID to set the session ID. Alternately you can pingAndSetSessionAuth if you are using this property with pingConnection. The pingAndSetSessionAuth property runs the ping and session-authorization commands in a single command batch, which may improve performance.	`[true|false]` Choose a value of 1 to use an ANSI SQL set session authorization command at the start of each database transaction. Set to 0 to use session-based authorizations.

Name	Description	Supported values
Set Session Auth System ID	If Set Session Authorization is enabled, specifies the database identity to use when the application server accesses the database from a transaction that runs with "system" identity.	`<database identity>` Replace <database identity> with the database identifier.
Start Wait	Sets the wait time (in seconds) before a connection problem is reported. If the start command completes successfully within this time period, no exceptions are reported in the server log. startWait time is used only with the databaseStartCommand property.	`<int>` Replace <int> with the number of seconds Unwired Server waits before reporting an error.
Truncate Nanoseconds	Sets a divisor/multiplier that is used to round the nanoseconds value in a java.sql.Timestamp to a granularity that the DBMS supports.	`[default|<int>]` The default is 10 000 000.
Use Quoted Identifiers	Specifies whether or not SQL identifiers are quoted.	`[True|False]` Set to true to enable use of quoted identifiers, or false to disable.
User	Identifies the user who is connecting to the database.	`[default|<user name>]` Replace <user name> with the database user name.
XA Datasource Class	Specifies the class name or library name used to support two-phase commit transactions, and the name of the XA resource library.	`<class name>` Replace <class name> with the class or library name. • SQL Anywhere database: `com.sybase.jdbc3.jdbc.SybXADataSource` • Oracle database: `oracle.jdbc.xa.client.OracleXADataSource`

SAP Java Connector Properties

Configure SAP Java Connector (JCo) connection properties.

For a comprehensive list of SAP JCo properties you can use to create an instance of a client connection to a remote SAP system, see *http://help.sap.com/javadocs/NW04/current/jc/com/sap/mw/jco/JCO.html#createClient(java.util.Properties)*.

Note: If Unwired Server is connecting to SAP with a Java connector, ensure you have copied required files to correct locations. See *Preparing Unwired Server to Connect to SAP* in *System Administration*.

Table 9. General connection parameters

Name	Description	Supported values
Client Number	Specifies the SAP client.	Three-digit client number; preserve leading zeros if they appear in the number
Logon User	Specifies the login user ID.	User name for logging in to the SAP system If using X.509 certificate authentication, remove the JCo properties `jco.client.passwd` and `jco.client.user` defined for the SAP connection profile in Sybase Control Center (SCC).
Password	Specifies the login password.	Password for logging in to the SAP system
Language	Specifies a login language.	ISO two-character language code (for example, EN, DE, FR), or SAP-specific single-character language code. As a result, only the first two characters are ever used, even if a longer string is entered. The default is EN.
System Number	Indicates the SAP system number.	SAP system number
Host Name	Identifies the SAP application server.	Host name of a specific SAP application server
Message Server	Identifies the SAP message server.	Host name of the message server

Name	Description	Supported values
Gateway Host	Identifies the SAP gateway host.	Host name of the SAP gateway Example: `GWHOST=hs0311`
Gateway Service	Identifies the SAP gateway service.	Service name of the SAP gateway Example: `GWSERV=sapgw53`
R/3 Name	Specifies R/3 name.	Name of the SAP system
Server Group	Identifies the group of SAP application servers.	Group name of the application servers
External Server Program	Identifies the program ID of the external server program.	Path and name of the external RFC server program, or program ID of a registered RFC server program Example: `TPNAME=/sap/srfcserv`
External Server Program Host	Identifies the host of the external server program. This information determines whether the RFC client connects to an RFC server started by the SAP gateway or to an already registered RFC server. **Note:** If the gateway host and external server program host are different, make sure that the SAP gateway has access to start the server program through a remote shell.	Host name of the external RFC server program Example: `TPHOST=hs0311`
Remote Host Type	Identifies the type of remote host.	2: R/2 3: R/3 E: external
RFC Trace	Specifies whether or not to enable RFC trace.	0: disable 1: enable

Name	Description	Supported values
Initial Codepage	Identifies the initial code page in SAP notation. A code page is used whenever character data is processed on the application server, appears on the front end, or is rendered by a printer.	Four-digit SAP code page number
Enable ABAP Debugging	Enables or disables ABAP debugging. If enabled, the connection is opened in debug mode and the invoked function module can be stepped through in the debugger. For debugging, an SAP graphical user interface (SAPGUI) must be installed on the same machine the client program is running on. This can be either a normal Windows SAPGUI or a Java GUI on Linux/UNIX systems.	0: no debugging 1: attach a visible SAPGUI and break at the first ABAP statement of the invoked function module
Remote GUI	Specifies whether a remote SAP graphical user interface (SAPGUI) should be attached to the connection. Some older BAPIs need an SAPGUI because they try to send screen output to the client while executing.	0: no SAPGUI 1: attach an "invisible" SAPGUI, which receives and ignores the screen output 2: attach a visible SAPGUI For values other than 0 a SAPGUI needs to be installed on the machine, where the client program is running. This can be either a Windows SAPGUI or a Java GUI on Linux/Unix systems.
Get SSO Ticket	Generates an SSO2 ticket for the user after login to allow single sign-on. If RfcOpenConnection() succeeds, you can retrieve the ticket with RfcGetPartnerSSOTicket() and use it for additional logins to systems supporting the same user base.	0: do not generate SSO2 ticket 1: generate SSO2 ticket

Name	Description	Supported values
Use Cookie Version 2	Indicates whether or not to use the specified SAP Cookie Version 2 as the login ticket instead of user ID and password.	User: $MYSAPSSO2$ Password: Base64-encoded ticket Login with single sign-on is based on secure network connection (SNC) encryption and can only be used in combination with an SNC.
Use X509	Indicates whether or not to use the specified X509 certificate as the login certificate instead of user ID and password.	User: $X509CERT$ Password: Base64-encoded ticket Login with X509 is based on secure network connection (SNC) encryption and can only be used in combination with an SNC.
Logon Check	Enables or disables login check at open time.	0: disable 1: enable If you set this to 0, RfcOpenConnection() opens a network connection, but does not perform the login procedure. Therefore, no user session is created inside the back-end system. This parameter is intended only for executing the function module RFC_PING.
Additional GUI Data	Provides additional data for graphical user interface (GUI) to specify the SAProuter connection data for the SAPGUI when it is used with RFC.	/H/ *router string* : the entire router string for the SAPGUI /P/ *password* : specify this value if the password for the SAPGUI connection is not the same as the password for the RFC connection.
GUI Redirect Host	Identifies which host to redirect the remote graphical user interface to.	Host name
GUI Redirect Service	Identifies which service to redirect the remote graphical user interface to.	Name of the service
Remote GUI Start Program	Indicates the program ID of the server that starts the remote graphical user interface.	Program ID of the server

Name	Description	Supported values
SNC Mode	Enables or disables secure network connection mode.	0: off 1: on
SNC Partner	Identifies the secure network connection partner.	Secure network connection name of the application server (for example, p:CN=R3, O=XYZ-INC, C=EN)
SNC Level	Specifies the secure network connection security level.	1: digital signature 2: digital signature and encryption 3: digital signature, encryption, and user authentication 8: default value defined by backend system 9: maximum value that the current security product supports
SNC Name	Indicates the secure network connection name. This property overrides the default secure network connection partner.	Token or identifier representing the external RFC program
SNC Service Lib Path	Identifies the path to the library that provides secure network connection service.	Full path and name of third-party security library
R/2 Destination	Identifies a configured R/2 system defined in the sideinfo configuration.	
Logon ID	Defines the string for SAPLOGON on 32-bit Windows.	String key to read parameters from the saplogon.ini file created by the SAPLogon GUI program on Windows
External Authentication Data	Provides data for external authentication (PAS). This is an old login mechanism similar to SSO; Sybase recommends that you do not use this approach.	
External Authentication	Specifies type of external authentication (PAS). See External Authentication Data property.	

SAP DOE-C Properties

Configure SAP Data Orchestration Engine Connector (DOE-C) properties. This type of connection is available in the list of connection templates only when you deploy a DOE-C package. No template exists for these types of connections.

Note: If you change the username or password property of a DOE-C connection, you must reopen the same dialog and click `Test Connection` after saving. Otherwise the error state of this DOE-C package is not set properly, and an error message is displayed. This will not work if you click `Test Connection` before saving the properties.

Name	Description	Supported values
Username	Specifies the SAP user account ID. The SAP user account is used during interaction between the connected SAP system and client for certain administrative activities, such as sending acknowledgment messages during day-to-day operations or "unsubscribe" messages if a subscription for this connection is removed. This account is not used for messages containing business data; those types of messages are always sent within the context of a session authenticated with credentials provided by the mobile client. The technical user name and password must be set to perform actions on subscriptions.	Valid SAP login name for the DOE host system.
Password	Specifies the password for the SAP user account.	Valid password.
DOE SOAP Timeout	Specifies a timeout window during which unresponsive DOE requests are aborted.	Positive value (in seconds). The default is 420 (7 minutes).

Name	Description	Supported values
DOE Extract Window	Specifies the number of messages allowed in the DOE extract window.	Positive value (in messages). The default is 50. When the number of messages in the DOE extract window reaches 50% of this value, DOE-C sends a `StatusReqFromClient` message, to advise the SAP DOE system of the client's messaging status and acknowledge the server's state. The default value is 50.
Packet Drop Size	Specifies the size, in bytes, of the largest JavaScript Object Notation (JSON) message that the DOE connector processes on behalf of a JSON client. The packet drop threshold size should be carefully chosen, so that it is larger than the largest message sent from the DOE to the client, but smaller than the maximum message size which may be processed by the client. Messages larger than the packet drop threshold size causes the subscription to enter the DOE packet drop state and become unusable.	Positive value (in bytes). The default is 1MB. Do not set higher than 2MB, or lower than 4096.
Service Address	Specifies the DOE URL.	Valid DOE URL. If you are using DOE-C with SSO: • Modify the port from the standard http://host:8000 to https://host:8001/. • Add the certificate being used as the technical user and DOE-C endpoint security profile certificate to the SAP DOE system's SSL Server certificate list by using the STRUST transaction. See your SAP documentation for details.

Name	Description	Supported values
Listener URL	Specifies the DOE-C server listener URL.	Valid DOE-C listener URL.
Security Profile	Specifies the security profile for the DOE-C endpoint.	Valid security profile.
SAP Technical User 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. If you are using DOE-C with SSO use the "SAP Technical User Certificate Alias" only for configurations which require the technical user to identify itself using an X.509 certificate; it specifies the Certificate Alias to be used as the technical user. This overrides the "Username" and "Password" settings normally used.	Valid certificate alias.

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

Name	Description	Supported values
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 store.

Device Users

Device users are individuals whose devices have been registered manually, through messaging-based applications, or automatically, through replication-based applications.

Device users are managed in Sybase Control Center (SCC) according to the device they use to synchronize data. The device user is a mechanism that identifies the person that controls the device. For messaging environments, the ID for this users consists of abbreviated user name. In some cases, ID of the device user can match the user name property that the user defines as part of the connection settings of the messaging client on the device.

The device users contrasts with the concept of an application user. An application user is the actual back-end EIS identity. The application user ID is the mechanism used to authenticate data request with the security provider you configure for Unwired Server. An application user ID is required when a person subscribes to a package, or when the user preforms replay operations back to the EIS.

Devices

Devices interact with Unwired Platform to gain access to corporate information. Registration of a device creates an account that identifies a user by the device registered. The user must authenticate with the authentication provider before any mobile business applications can be accessed from the device application.

Devices are categorized by the synchronization model used by the device application. The type of synchronization model used determines the configuration and administration actions you can perform.

Device Information

Access device information in the Devices tab of the Device Users node. View data on registered devices in order to manage and monitor device synchronization.

Select one of the following modes:

- **Unified** – lists all registered replication- and messaging-based synchronization devices. This view presents key information including the current registration status of the device,

the device synchronization mode, and when the device last connected with Unwired Server.

- **RBS** – lists replication-based synchronization devices that are automatically registered when replication-based application users synchronize successfully with the server.
- **MBS** – lists messaging-based synchronization devices that are currently registered either manually or using the public API (for bulk registration). This view allows the administrator to see messaging device status information that is useful for diagnostic purposes.

View the following device information, depending on the filter you choose:

Property	Description	Filter
Device ID	The unique identifier attached to the device hardware of a registered device. In MBS mode, until the device is registered, the device ID remains empty string in MBS mode. In Unified or RBS mode, the device ID is `MBS_<number>`. However, the permanent ID is not assigned until the device connects to Unwired Server. For example, after Windows Mobile 6 simulator connects to the Unwired Server, the assigns an permanent ID as `Emulator324567336`.	All
Device Platform	The operating platform that the device uses.	All
Device Type	The type of device. For example, if the device model is a BlackBerry, the type is the form factor (for example, BlackBerry Bold).	All
Registered Date	The date of initial device registration.	All
Last Connected	The time at which the last communication took place between Unwired Server and the device.	All
RBS Status	The time at which the last communication took place between Unwired Server and the device.	Unified and MBS

Property	Description	Filter
MBS Status	The registration status of the device for messaging-based synchronization.	Unified and RBS
Lock Status	Locked or Unlocked. Synchronization is disabled in locked devices	All
Activation User Name	Tthe name of the user that activates the device.	MBS
Status	The current status of a device. The possible values are: Running, Suspended, Pending Activation, Online, Offline, and Expired. **Note:** If the device connection is through Relay Server, the connection to Unwired Server remains open for up to 6 minutes after the device has dropped its connection to Relay Server. Thus, the Status column may incorrectly show that the device is online for up to 6 minutes after the device has disconnected.	MBS
Pending Items	The pending items on the server side that needs to be sent to the device. The device must have connect to server at least once, before information in this column appears.	MBS
Activation Code Expire	The date the activation code expires on.	MBS
Last Delivery	The date and time of last item delivered from Unwired Server to the device. The device must have connect to server at least once, before information in this column appears.	MBS
Client Version	The Sybase Unwired Platform client runtime version of the client. The device must have connect to server at least once, before information in this column appears	MBS

Property	Description	Filter
Node	The cluster node on which the device is registered.	MBS

These columns can be used to sort the data by clicking the column title.

Advanced MBS Properties

If the device type is MBS, toggle **Advanced** to show or hide these columns:

Property	Description
Registered Date	The date the device was registered.
Last Connected	The date and time the device last connected.
RBS Status	Whether the MBS device is also used for RBS synchronization.
Lock Status	Whether the device has been locked.

These columns cannot be used to sort data.

Searching for Devices

Set search criteria to filter devices viewed in the Devices tab.

1. In the left navigation pane, click **Device Users**.
2. In the right administration pane, click the **Devices** tab.
3. To set the search criteria, configure these search elements:
   a) Choose the device information column name you want to enter a search value for.

      Note: MBS devices cannot search on columns that display date or time information.

   b) Type or choose the value for the column name you selected.

Replication Devices

Replication devices are used with replication-based synchronization (RBS) mobile business objects that rely on RBS data cached in the consolidated database. RBS device users are automatically registered when they first synchronize data. There is no device configuration required; the only tasks an administrator performs are monitoring RBS device activity, locking and unlocking RBS devices, and deleting them.

An administrator can lock or unlock devices to disallow or allow users from the device to access the Unwired cluster. All activities, including sending of the push notifications, are stopped while a device is locked. The device application will get an error when trying to communicate with Unwired Server. See *Sybase Control Center > Manage > Managing Unwired Platform > Routine Command and Control Actions > Provision > Device Users > Devices > Locking and Unlocking Devices*.

Messaging Devices

Messaging devices contain applications that send and receive data through messaging. An platform administrator must configure the device activation template properties for messaging-based synchronization (MBS) devices. Device activation requires user registration. Upon successful registration, the device is activated and set up with the template the administrator has selected.

Device registration pairs a user and a device once the user supplies the correct activation code. This information is stored in the messaging database, which contains extensive information about users and their corresponding mobile devices.

Users who are registered but who have not yet installed the software are listed in the window as **registered**, and their messages are queued by Unwired Server for later delivery.

Typically, device registration occurs when the user initially attempts to connect to Unwired Server. However, an administrator can force a user to reregister if there is data corruption on the device, or if the user is assigned a new device. This reestablishes the relationship between the user and the device and refreshes the entire data set on the device.

Note: The device user must activate the messaging account within the number of hours specified in their activation message. If a user does not activate the account within that time frame, an administrator must reregister the user.

A platform administrator can also lock or unlock devices to disallow or allow users from the device to access the Unwired cluster. All exchanges between the device and Unwired Server are stopped while a device is locked. The device application will get an error when trying to communicate with Unwired Server. See *Sybase Control Center > Manage > Managing Unwired Platform > Routine Command and Control Actions > Provision > Device Users > Devices Locking and Unlocking Devices*.

Registering and Reregistering Messaging Devices

Use Sybase Control Center to trigger the registration and device activation process, which allows messaging mobile business objects (MBOs) to handle messages belonging to different data sources.

Note: When using a Windows Mobile emulator or BlackBerry simulator to register a device in Sybase Control Center, the device ID changes each time you reset the emulator to factory settings and reinstall the client. Before reinstalling, you must delete the original device from Unwired Server. Then, reregister the device. Otherwise, the device log shows a `Wrong Device for Code` error when the device attempts to connect after registration. This problem occurs with Windows Mobile emulator and BlackBerry simulator devices.

1. In the left navigation pane, click the **Device Users** node.
2. In the right administration pane, click the **Devices** tab.

3. Click **Register** to register a new device, or **Reregister** to update the device used of an existing device user.
4. In the **Register Device** or the **Reregister Device** dialog:
   a) For new device registrations 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 device registration. If you have not created any templates, only **Default** appears in the list.
   The template you choose supplies initial values in the subsequent device activation fields.
5. Change the default activation field values for the template you have chosen. If you are using the default template, you must provide the server name, which is empty.

   If you are using a 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 letters A – Z (uppercase or lowercase), numbers 0 – 9, or a combination of both. Default: 0.

     Note: If the device uses relay server to connect to Unwired Server, the farm ID should be the name of the Unwired Server farm configured in the relay server for messaging-based synchronization applications. If the device connects to Unwired Server directly, the farm ID should be 0.
   - **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 **Activation Code** to enter the code sent to the user in the activation e-mail. This value can contain letters A – Z (uppercase or lowercase), numbers 0 – 9, or a combination of both. Acceptable range: 1 to 10 characters.

   If the activation code is automatically generated, the code for the device can be retrieved from the **Connections** group of the **Device Properties** dialog
7. Click **OK**.

Assigning and Unassigning Workflow Packages to Device Users

Assign mobile workflow packages to messaging devices make them available to an activation user. You can also unassign mobile workflow packages at any time.

1. In the left navigation pane, click **ClusterName** > **Device Users** > **Devices** tab.

2. In the right administration pane, click the **Devices** tab, select an MBS device, then click **Workflows**.
3. Select the workflow packages you want to assign or unassign.

   The Workflow Assignment dialog shows workflows that are already assigned to the selected MBS device. Select one of the workflows, then click **Unassign workflows** to unassign, or **Assign workflows** to assign it to another workflow, in the Assign Workflows dialog.
4. In the confirmation dialog, click OK.

Messaging Device Connection Properties

Connection properties define the connection information used by Unwired Server to relate a user to a device.

- **Activation Code –** 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.

Messaging Device 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 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.

Messaging 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. Acceptable values:
  - For IIS – use`/ias_relay_server/client/rs_client.dll`.
  - For Apache – use `/cli/iasrelayserver`.

- **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.
- **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.

Messaging Device Information 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.

Messaging Device Features Properties

Features properties indicate whether mobile workflow activity is enabled for the messaging device.

- **Workflows Enabled –** defines whether an administrator can assign mobile workflows to the messaging device. Accepted values: true (default) or false.

Messaging Device Scheduled Sync Properties

Scheduled sync properties determine when and how frequently messaging-based synchronization occurs. These settings apply to Windows Mobile devices only.

- **Enabled –** enables scheduled push synchronization to the device if the device is online. This feature uses a messaging-based push for Unwired Server synchronization. Unlike an IP push, which maintains a persistent IP connection, a scheduled sync push uses an IP connection only long enough for the data exchange to complete according to peak and off-peak device usage frequency. The feature overcomes network issues with always-on connectivity and battery life consumption on 3G networks. Acceptable values: true (enabled) or false (disabled). Default: false.

- **Peak Days –** the days of the week the device is most frequently used, listed in a comma-separated list. Acceptable values: Mon, Tues, Wed, Thurs, Fri, Sat, Sunday. Default: Mon, Tues, Wed, Thurs, Fri.
- **Peak End Time –** the time of day at which peak hours end on peak days. Acceptable values: any 24-hour clock entries, specified as HH:MM. Default: 18:00.
- **Peak Start Time –** the time of day at which peak hours begin on peak days. Acceptable values: any 24-hour clock entries, specified as HH:MM. Default: 8:00.
- **Off-Peak Frequency –** the frequency of scheduled sync during off-peak times. Acceptable values: 5 minutes, 10 minutes, 15 minutes, 30 minutes, 1 hour, 2 hours, 4 hours, As Items Arrive, or Manual. Selecting the "As Items Arrive" option is equivalent to operating in IP push mode. Default: 1 hour.
- **Peak Frequency –** the frequency of scheduled sync during peak times on peak days. Acceptable values: 5 minutes, 10 minutes, 15 minutes, 30 minutes, 1 hour, 2 hours, 4 hours, As Items Arrive, or Manual. Default: 15 minutes.

Example

Consider hospital employees that work different peak hours, depending on the shift they work, and have a Patient MBO that updates staff of changes in the health and treatment of each patient. The administrator may have multiple templates for each shift group, as well as the role each group performs at the hospital. For example, nursing staff may not require frequent updates during off-peak hours when they are away from the hospital and would set the off-peak frequency at 1 hour. However, the nursing staff would require almost immediate updates while on shift, as frequently as they happen. Surgical staff or "on call" staff may require frequent updates even when they are not on shift at the hospital, wherein administrators, may keep the peak frequency at "As Items Arrive", but also set the off-peak frequency to "As Items Arrive" to ensure patient information remains up-to-date on the device.

Messaging Device User Registration Properties

Device user registration properties allow you to customize the registration request that is delivered to the device.

- **Activation Code Expiration –** defines how long a user has to activate their account, in hours, before the account activation period expires. Acceptable values: 1 to 10,000 hours. Default: 72 hours.
- **Activation Code Length –** the number of characters to be contained in the activation code. Acceptable values: 1 to 10. Default: 3.

Messaging Device Apple Push Notification Properties

Apple push notification properties allow iPhone users to install messaging 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 iPhone OS 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 iPhone 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 iPhone OS system-sound facility, they must be in one of the supported audio data formats. See the iPhone 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 iPhone OS 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

Messaging Device BlackBerry Notification Properties

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

Property	Description
Enabled	Enables notifications to the device if the device is offline. This feature uses an SMS-based push notification for the Send/Receive data exchange over an IP connection only long enough to complete the Send/Receive data exchange. The feature overcomes network issues with always-on connectivity and battery life consumption on 3G 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

Property	Description
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
Name	The BES server to which this device's notifications are sent. In cases where there there are multiple BES servers in an organization, define all BES servers.

Device Templates

A device template is a property sheet that contains default messaging device settings for user registration. There are two types of template: default and custom.

Installing Unwired Platform automatically creates a single default template. You can modify and use the default template as your standard template. You can also design custom templates to meet different requirements of users, managers, and other groups within your organization. Before you register users, consider whether you plan to use the default template or create one or more custom templates.

After you use templates to configure device settings at registration, you can make subsequent changes to the template. These changes do not affect any users currently registered with that template. For example, once a device is provisioned using a Manager template, any later changes to the Manager template do not affect the user's individual settings.

Configuring Device Templates

Modify and use the default template as your standard messaging device template, or create one or more custom templates to better meet the different requirements of users or groups within your organization.

1. In the left navigation pane, click the **Device Users** node.
2. In the right administration pane, click the **Device Templates** tab.
3. Do one of the following:
   - To modify and use an existing template, select the template name from the list of available templates and click **Properties**.
   - To create a new template, click **New**.
4. In the Template dialog, configure these property categories as required:
   - Apple Push Notifications

- Connection
- Custom Settings
- Device Advanced
- Device Info
- Features
- Scheduled Sync
- User Registration

5. Click **OK**.

Messaging Device Apple Push Notification Properties

Apple push notification properties allow iPhone users to install messaging 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 iPhone OS 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 iPhone 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 iPhone OS system-sound facility, they must be in one of the supported audio data formats. See the iPhone 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 iPhone OS 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

Messaging Device Connection Properties

Connection properties define the connection information used by Unwired Server to relate a user to a device.

- **Activation Code –** 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.

Messaging Device 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 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.

Messaging 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. Acceptable values:
  - For IIS – use`/ias_relay_server/client/rs_client.dll`.
  - For Apache – use `/cli/iasrelayserver`.
- **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.
- **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.

Messaging Device Information 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.

Messaging Device Features Properties

Features properties indicate whether mobile workflow activity is enabled for the messaging device.

- **Workflows Enabled –** defines whether an administrator can assign mobile workflows to the messaging device. Accepted values: true (default) or false.

Messaging Device Scheduled Sync Properties

Scheduled sync properties determine when and how frequently messaging-based synchronization occurs. These settings apply to Windows Mobile devices only.

- **Enabled –** enables scheduled push synchronization to the device if the device is online. This feature uses a messaging-based push for Unwired Server synchronization. Unlike an IP push, which maintains a persistent IP connection, a scheduled sync push uses an IP connection only long enough for the data exchange to complete according to peak and off-peak device usage frequency. The feature overcomes network issues with always-on connectivity and battery life consumption on 3G networks. Acceptable values: true (enabled) or false (disabled). Default: false.
- **Peak Days –** the days of the week the device is most frequently used, listed in a comma-separated list. Acceptable values: Mon, Tues, Wed, Thurs, Fri, Sat, Sunday. Default: Mon, Tues, Wed, Thurs, Fri.
- **Peak End Time –** the time of day at which peak hours end on peak days. Acceptable values: any 24-hour clock entries, specified as HH:MM. Default: 18:00.
- **Peak Start Time –** the time of day at which peak hours begin on peak days. Acceptable values: any 24-hour clock entries, specified as HH:MM. Default: 8:00.
- **Off-Peak Frequency –** the frequency of scheduled sync during off-peak times. Acceptable values: 5 minutes, 10 minutes, 15 minutes, 30 minutes, 1 hour, 2 hours, 4

hours, As Items Arrive, or Manual. Selecting the "As Items Arrive" option is equivalent to operating in IP push mode. Default: 1 hour.

- **Peak Frequency –** the frequency of scheduled sync during peak times on peak days. Acceptable values: 5 minutes, 10 minutes, 15 minutes, 30 minutes, 1 hour, 2 hours, 4 hours, As Items Arrive, or Manual. Default: 15 minutes.

Example

Consider hospital employees that work different peak hours, depending on the shift they work, and have a Patient MBO that updates staff of changes in the health and treatment of each patient. The administrator may have multiple templates for each shift group, as well as the role each group performs at the hospital. For example, nursing staff may not require frequent updates during off-peak hours when they are away from the hospital and would set the off-peak frequency at 1 hour. However, the nursing staff would require almost immediate updates while on shift, as frequently as they happen. Surgical staff or "on call" staff may require frequent updates even when they are not on shift at the hospital, wherein administrators, may keep the peak frequency at "As Items Arrive", but also set the off-peak frequency to "As Items Arrive" to ensure patient information remains up-to-date on the device.

Messaging Device User Registration Properties

Device user registration properties allow you to customize the registration request that is delivered to the device.

- **Activation Code Expiration –** defines how long a user has to activate their account, in hours, before the account activation period expires. Acceptable values: 1 to 10,000 hours. Default: 72 hours.
- **Activation Code Length –** the number of characters to be contained in the activation code. Acceptable values: 1 to 10. Default: 3.

Users

In Unwired Platform, application users are individuals who have been registered through messaging-based or replication-based applications. Application users are managed in Sybase Control Center.

A user is automatically registered upon first successful authentication by the security provider associated with the package that user is trying to access from a device. A user can have multiple devices.

The platform administrator can view the user name and the security configuration that was used to authenticate the user. The administrator can also review devices associated with a user to perform any device deletion to free up license. 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 Users

Delete a user to remove the entry as well as personalization data from the consolidated database.

1. In the left navigation pane of Sybase Control Center, click the **Users** node and select the **Users** tab in the right administration pane.
2. Select the user and click **Delete**.

The user entry and data are removed.

Checking User Device Assignments

Check which devices are used by registered users.

1. In the left navigation pane of Sybase Control Center, click the **Users** node and select the **Users** tab in the right administration pane.
2. Select the user and click **Devices**.

All devices used by the user are listed in the dialog.

Security Configurations

A security configuration determines the scope of user identity, authentication and authorization checks, and can be assigned to one or more domains for both messaging and replication-based applications. For example, user "John" may be authenticated one way in one security configuration, but authenticated differently another security configuration, even when "John" represent the same human user.

A security configuration aggregates various security mechanisms for protecting Unwired Platform resources under a specific name, which administrators can then assign to domains. Each security configuration consists of:

- A set of configured security providers
- Security roles
- Role mappings (set at the domain and package level)

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 to access a data set 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 policy 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, expand the **Security** folder.
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 to set a authentication cache timeout value.

   The timeout determines how long authentication results should be cached before a user is required to reauthenticate. For details, see *Authentication Cache Timeouts* in *System Administration*. To configure this value:

   a) Set the cache timeout value in seconds. The default is 3600.
   b) Click **Save**.
6. Select the tab corresponding to the type of security provider you want to configure: Authentication, Authorization, Attribution, or Audit.
7. 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**.
8. 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.

      Note: If you are using SAPSSOTokenLoginModule, change the "token expiration interval" property so the value is be larger than the value of the cache timeout property.
   d) Click **OK**.
      The configuration is saved locally, but not yet committed to the server.
9. 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.

10. 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, attribution, 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-XX\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.
- Attribution modules – when a user is authenticated, the attribution provider adds more information about the authenticated user. Attribution modules are optional; however, if the user is providing registration information, Sybase recommends that you use this module.
- Auditing modules – report all audit events to allow you to evaluate the security system implementation for Unwired Server. Auditing provides you a record of all the security decisions that have been made. Each successful authentication creates a session key that shows up in subsequent security checks for that user. Unsuccessful authentications are also logged. Each authorization records what roles were checked, or what resource was accessed. Audit filters determine what events get recorded, the audit format determines what the audit records look like, and the audit destination specifies where audit records are sent. Use the audit trail to identify who did what and when, with respect to objects secured by your providers. Auditing modules are optional.

In most cases, each security module requires a unique set of configuration properties. However, there are some cases when modules require a common set of properties, and these properties are configured once for each module on a tab created for that purpose.

For more information on configuring security providers, see *Sybase Unwired Platform System Administration Guide > Security Administration > Implementing System Wide Security > User Security Setup*.

No Security Provider

A NoSec provider offers pass-through security for Unwired Server, which is intended for use in development environments or for deployments that require no security control.

If you use NoSec providers, all login attempts succeed, no matter what values are used for the user name and password. Additionally, all role and control checks based on attributes also succeed.

Sybase has used these classes to implement the NoSec provider:

- **NoSecLoginModule** – provides pass-through authentication services.
- **NoSecAttributer** – provides pass-through attribution services.
- **NoSecAuthorizer** – provides pass-through authorization services.

Disabling Security with the NoSec Provider

Disable security using the NoSec provider.

1. In the left navigation pane, expand the **Security** folder.
2. In the right administration pane, click **New**.
3. Enter a name for the No 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 **com.sybase.security.core.NoSec<*ModuleType*>** as the security provider for authentication and attribution. Leave both the authorization and audit security providers empty.
   a) Select the tab corresponding to the appropriate security type.
   b) Click **New**.
   c) From the menu, select **com.sybase.security.core.NoSec<*ModuleType*>** .
   d) Configure the properties associated with the provider by setting values according to your requirements. Add properties as required. For more information about configuring NoSec security provider properties, see the individual reference topic.
   e) Click **OK**.
6. 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.
7. 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.

Next

Assign the newly created security configuration to the domain where the package that requires no security is deployed.

Stacking Multiple Security Providers

Stack multiple providers to provide a security solution that meets more complex security requirements.

Each provider has a controlFlag attribute that controls overall behavior when you enable two or more providers.

1. Use Sybase Control Center to create a security configuration and add multiple providers as required for authentication, authorization, attribution, and audit. For details, see *Sybase Control Center > Configure > Configure Unwired Platform > Security*.
2. 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.
3. Click **Save**.
4. Select the **General** tab, and click **Apply**.

Stacking LoginModules in SSO Configurations

Use loginmodule stacking to enable role-based authorization for MBOs and data change notification (DCN).

Stacking LDAPLoginModule, LDAPAttributer, and SAPSSOTokenLoginModule to enable role-based authorization with SSO

Neither the SAPSSOTokenLoginModule or the CertificateAuthenticationLoginModule login modules extract role information. If MBOs and MBO operations have roles assigned, stack login modules to get roles for the user, using one of these methods:

1. If SAP is configured to use LDAP/Active Directory as JAAS providers within its Java stack for granting an SSO2 token, configure a stacked LDAPLoginModule pointing to the same LDAP/Active Directory to separately authenticate and retrieve roles. This method assumes the user name and password credentials are authenticated by those modules as well.
2. Rely on the "csi-userrole" provider.

See *Configuring an LDAP Authentication Module* in Sybase Control Center online help.

Stacking modules to enable DCN with SSO

Stack multiple LoginModules with an appropriate set of controlFlag settings to enable DCN in the same SSO enabled package. All DCN operations require the "DCNUser" logical role in the named security configuration (role mapping applies). An additional LoginModule with authorization is required that can assign a physical role. The stacking of modules authenticates

DCN users, and grants them the DCN role. Ordering of modules and control flag settings in the security configuration can vary. For example:

1. The SAPSSOTokenLoginModule is first in the list with the control flag set to "sufficient". If authentication succeeds, none of the other Login Modules are called unless their control flags are set to "Required".
2. A Login Module for DCN users that is also "sufficient", and is paired with a module that retrieves roles.

See *Stacking Multiple Security Providers* .

controlFlag Attribute Values

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 to succeed. Authentication proceeds down the LoginModule list.
requisite	The LoginModule is required to succeed. 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 to succeed. 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 to succeed. Irrespective of success or failure, authentication proceeds down the LoginModule list.

Example

Say you list providers in the following order and set the corresponding controlFlag attributes as follows:

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, then they go to LDAP (first), 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.

LDAP Module Stacking Considerations

LDAP providers can sometimes share a common configuration.

If you want to share the configuration between the login provider and the attribution provider, set the configuration for the login provider but do not specify values for the attribution provider. In this case, the attribution provider will automatically inherit the appropriate configuration from the login provider.

Note: You cannot share configuration between the login provider and the authorization provider. The authorization provider always expects its own configuration.

Other considerations include:

- The attribution provider has maximum functionality when combined with the LDAP authentication provider; however, the attribution provider can be configured completely standalone or with alternate authentication providers.
- If you do not configure a login provider, you must define the configure all properties in the attribution provider instead.
- A complete configuration defined for the attribution provider override that of a login module. While it is possible to have separate configurations for the authentication and attribution providers, Sybase does not recommend this setup.

Reordering Configured Providers

List stacked security providers for a security configuration to identify them as primary or auxiliary providers. Authentication, authorization, or attribution 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, Attribution, or Audit.
4. Select a provider from the list, then use the up and down arrows to the right of the table to achieve the desired placement.

5. 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, authorization, audit, or attribution.

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.

NoSec Configuration Properties

A NoSec provider offers pass-through security for Unwired Server. You can apply a NoSec security provider for authentication, authorization, and attribution in the Security node of Sybase Control Center.

The NoSecLoginModule class provides open authentication services, the NoSecAuthorizer class provides authorization services, and the NoSecAttributer provides attribution services. However, you need to configure only authentication properties for a NoSec provider.

Table 10. Authentication properties

Property	Default Value	Description
useUsernameAsIdentity	true	If this option is set to true, the user name supplied in the call-back 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.

LDAP Configuration Properties

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.

- The **LDAPLoginModule** provides authentication services. Through appropriate configuration, you can enable certificate authentication in LDAPLoginModule.
- Optional. The **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 always used with every security configuration; however it is not displayed in Sybase Control Center.

  Only use LDAPAuthorizer 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 configure properties for it explicitly.

It cannot share the configuration options specified for the LDAPLoginModule (if any are configured).

- Optional. The **LDAPAttributer** provides attribution services. The attribution provider can share configuration defined for the LDAPLoginModule. That is, if no explicit configuration properties are specified for LDAPAttributer, it uses the configuration information from the LDAPLoginModule, when only one LDAPLoginModule is configured. If there are more than one, then LDAPAttibuters cannot share properties, because they would not know which LoginModule to share with; in this case you must also configure propertes for each LDAPAttributer.

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

Property	Default Value	Description
ProviderURL	`ldap://local-host: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). Note that Development Editions of Unwired Platform include an OpenDS LDAP server that runs on a nonstandard port of 10389. However, most LDAP servers use the standard port of 389. Otherwise, use this syntax for setting the value: `ldap://<hostname>:<port>`
DefaultSearchBase	None	The LDAP search base that is used if no other search base is specified for authentication, roles, attribution and self registration: 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 "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.

Property	Default Value	Description
AuthenticationMethod	simple	The authentication method to use for all authentication requests into LDAP. Legal values are generally the same as those of the java.naming.security.authentication JNDI property. Choose one of: • simple — For clear-text password authentication. • DIGEST-MD5 — For more secure hashed password authentication. This method requires that the server use plain text password storage and only works with JRE 1.4 or later. See the*Java Sun* Web site for more information.
AuthenticationFilter	For most LDAP servers: `(&(uid={uid})(objectclass=person))` or For Active Directory email lookups: `(&(userPrincipalName={uid})(objectclass=user))[ActiveDirectory]` For Active Directory Windows username lookups: `(&(sAMAccountName={uid})(objectclass=user))`	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
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 value is not specified, the LDAP DefaultSearchBase 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. If you use an encrypted the password using the CSI encryption utility, append .e to the property name. For example: `CSI.loginModule.7.options.BindPassword.e=1-AAAAEgQQOLL+LpXJO8fO9T4SrQYRC9lRT1w5ePfdczQTDsP8iACk9mDAbm3F3p5a1wXWKK8+NdJuknc7w2nw5aGJlyG3xQ==`
RoleSearchBase	none	The search base used to retrieve lists of roles. If this value is not specified, the LDAP DefaultSearchBase is used.
RoleFilter	For SunONE/iPlanet: `(&(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 or this property is not initialized, no roles are available.

Property	Default Value	Description
RoleMemberAttributes	For Netscape Directory Server: member,uniquemember	The role's member attributes defines a comma-delimited list of attributes that roles may have that define a list of DN's of people who are in the 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 for retrieved roles that is the common name of the role. If this value is "dn" it is interpreted specially as the entire dn of the role as the role name.
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.
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.
UserFreeformRoleMembershipAttributes	None	The "freeform" role membership attribute list. Users who have attributes in this comma-delimited list are automatically granted access to roles whose names are equal to the attribute value. For example, if the value of this property is "department" and user's LDAP record has the following values for the department attribute, { "sales", "consulting" }, then the user will be granted roles whose names are "sales" and "consulting".

Property	Default Value	Description
Referral	ignore	The behavior when a referral is encountered. The valid values are those dictated by LdapContext, for example, "follow", "ignore", "throw".
DigestMD5Authentication-Format	DN For OpenLDAP: Username	The DIGEST-MD5 bind authentication identity format.
UseUserAccountControlAttribute	For most LDAP servers: false For ActiveDirectory: true	The UserAccountControl attribute to be 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	Indicates whether authentication with this login module is sufficient to allow the user to log in, or whether the user must also be authenticated with another login module. Rarely set to anything other than "sufficient" for any login module. **Note:** controlFlag is a generic login module option rather than an LDAP configuration property.

Certificate Configuration Properties

The certificate validation provider contributes only authentication services. For Unwired Server security, configure these properties from the corresponding tab in the Security node of Sybase Control Center.

To configure certificate validation with another security provider, ensure you configure the certificate validation properties before other login modules that will support this validation service.

Table 11. Authentication properties

Property	Default value	Description
validatedCertificateIsIdentity	false	Specifies if the certificate should be set as the ID for the authenticated subject. This option should be set to false if the `CertificateValidationLoginModule` is used in conjunction with other login modules that establish user identity based on the validated certificate.

NTProxy Configuration Properties

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 12. 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.

SAP SSO Token Authentication Properties

Add and configure authentication provider properties for SAPSSOTokenLoginModule or accept the default values

Table 13. 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.
SAP server URL	(Required) – the SAP server URL that authenticates the user and from which Unwired Server gets the SSO2 token.
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*. **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.

Certificate Authentication Properties

Add and configure authentication provider properties for CertificateAuthenticationLoginModule, or accept the default settings.

Table 14. 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.
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. Revoked CA certificates result in authentication failure.

Property	Description
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 `<Unwired-Platform_InstallDir\Servers\Unwired-Server\Repository\Securitytruststore\truststore.jks`. **Note:** This property is required only if Validate certificate path is set to true.
Trusted certificate store password	(Optional) The password required to access the trusted certificate store. For example, import the issuer of the certificate you are trying to authenticate into the shared JDK cacerts file and specify the password using this property. **Note:** This property is required only if Validate certificate path is set to true.
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.
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.

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 example: 1. Import a CA certificate (ca.crt) that is used to issue the client certificate (sybase101.p12) into truststore.jks: `keytool -import -keystore <UnwiredPlatform_installDir>\UnwiredPlatform\Servers\UnwiredServer\Repository\Security\truststore.jks -file <UnwiredPlatform_installDir>\UnwiredPlatform\Servers\UnwiredServer\Repository\Security\ca.crt` 2. Set these values for these properties: • Validate certificate path = `True` • Trusted certificate store = `<UnwiredPlatform_installDir>\UnwiredPlatform\Servers\UnwiredServer\Repository\Security\truststore.jks>` • Trusted certificate store password = `changeit` • Trusted certificate store provider = `SunJCE` • Trusted certificate store type = `JCEKS`

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.

Assigning a Security Configuration to a Domain

Assign security configurations to one or more domains. This allows the supAdmin to offer a security repository for application user authentication and authorization, as well as to share security providers across domains in case one tenant uses mutiple domains.

Prerequisites

A supAdmin must already have created one or more security configurations in the Security node of Sybase Control Center.

Task

1. In the left navigation pane, select the **Domains** folder and choose the domain you want to configure.
2. Select the **Security Configurations** tab and click **Assign**.
3. From the list of available security configurations, select the appropriate configuration for domain security, and click **OK**.
   If successful, an Assigned successfully message appears, and the newly added security configuration is listed in the domain-level Security node.

Mobile Workflows

Mobile Workflow packages support occasionally connected users and solve the replication and synchronization issues such users present with respect to data concurrency. Worfkflow

packages are similar to other package types in that an administrator must deploy the package to Unwired Server so that it can be configured and made available to client devices.

Workflow Devices Information

Access workflow device information in the Devices tab of the Workflows node. View data on registered devices in order to manage and monitor device synchronization.

View the following device information:

Property	Description
Activation User Name	The name of the user associated with the device ID. In MBS mode, until the device is registered, the device ID remains empty string in MBS mode. In Unified or RBS mode, the device ID is `MBS_<number>`. However, the permanent ID is not assigned until the device connects to Unwired Server. For example, after Windows Mobile 6 simulator connects to the Unwired Server, the assigns an permanent ID as `Emulator324567336`.
Device Type	The type of device. For example, if the device model is a BlackBerry, the type is the form factor (for example, BlackBerry Bold).
Device ID	The unique identifier attached to the device hardware of a registered device.
Status	The current status of a device. The possible values are: Running, Suspended, Pending Activation, Online, Offline, and Expired. **Note:** If the device connection is through Relay Server, the connection to Unwired Server remains open for up to 6 minutes after the device has dropped its connection to Relay Server. Thus, the Status column may incorrectly show that the device is online for up to 6 minutes after the device has disconnected.
Pending Items	The pending items on the server side that needs to be sent to the device.
Last Delivery	The date and time of last item delivered from Unwired Server to the device.
Client Version	The Sybase Unwired Platform client runtime version of the client.
Node	The cluster node on which the device is registered.

Searching for Workflow Devices

Set search criteria to filter devices viewed in the Devices tab for Workflows.

1. In the left navigation pane, click **Workflows**.
2. In the right administration pane, click the **Devices** tab.
3. To set the search criteria, configure these search elements:
   a) Choose the device information column name you want to enter a search value for.

      Note: Workflow devices cannot search on columns that display date or time information.

   b) Type or choose the value for the column name you selected.

Enabling and Configuring the Notification Mailbox

Configure the notification mailbox settings that allow Unwired Server to transform e-mail messages into mobile workflows.

The notification mailbox configuration uses a listener to scan all incoming e-mail messages delivered to the particular inbox specified during configuration. When the listener identifies an e-mail message that matches the rules specified by the administrator, it sends the message as a mobile workflow to the device that matches the rule.

Note: Saving changes to the notification mailbox configuration deletes all e-mail messages from the account. Before proceeding with configuration changes, consult your e-mail administrator if you want to back up the existing messages in the configured account.

1. In the left navigation pane, click **Workflows**.
2. In the right administration pane, click **Notification Mailbox**.
3. Select **Enable**.
4. Configure these properties:

   - **Protocol** – choose between POP3 or IMAP, depending on the e-mail server used.
   - **Use SSL** – encrypt the connection between Unwired Server and the e-mail server in your environment.
   - **Server** and **Port** – configure these connection properties so Unwired Server can connect to the e-mail server in your environment. The defaults are localhost and port 110 (unencrypted) or 995 (encrypted).
   - **User name** and **Password** – configure these login properties so Unwired Server can log in with a valid e-mail user identity.
   - **Truncation limit** – specify the maximum number of characters taken from the body text of the original e-mail message, and downloaded to the client during synchronization. If the body exceeds this number of characters, the listener truncates the body text to the number of specified characters before distributing it. The default is 5000 characters.

- **Poll seconds** – the number of seconds the listener sleeps between polls. During each poll, the listener checks the master inbox for new e-mail messages to process. The default is 60 seconds.

5. If you have added at least one distribution rule, you can click **Test** to test your configuration. If the test is successful, click **Save**.

Configuring a Mobile Workflow Package

Configure mobile workflow properties for you production environment.

Prerequisites

You must deploy a package before you can configure properties for it.

Configuring General Mobile Workflow Properties

Configure general properties for a mobile workflow, including display name and icon. Alter these settings to modify development environment values to production environment equivalents.

1. In the left navigation pane, click **Workflows** and select the mobile workflow for which to configure the properties.
2. In the right administration tab, click **General**.

   Only an administrator can change these properties. All others are configured by the mobile workflow developer and cannot be modified.

   - **Display name** – sets the name that appears for the mobile workflow package.
   - **Display icon** – scroll and select the icon you want to use for the mobile workflow package.
3. Click Lock/Unlock to lock or unlock a mobile workflow.

   A locked mobile workflow cannot be modified or deployed.
4. Click **Save**.

Configuring Matching Rules

Define the parameters and matching rules that determine how e-mail messages are redirected at runtime.

Prerequisites

The developer must have created an object query and added an E-mail Subscription starting point when the mobile workflow was designed.

Task

You can configure a matching rule at one of two levels:

- At the inbox level to route e-mails for all mobile workflows

- At the package level to route e-mails only for a specific mobile workflow

1. In the left navigation pane, click **Workflows** > ***WorkFlowName***.
2. In the right administration pane, click the **Matching Rules** tab.
3. Configure matching rules by either:
   - Clicking **Add** to create a new rule, or,
   - Selecting an existing rule name and clicking **Properties**.
4. In the **Matching Rules** dialog:
   a) Select the field in the e-mail from which the parameter value is extracted. For example, if you choose From, the parameter value is extracted from the line of the e-mail message that indicates the name of the sender of the message.

      You can also select one of the custom parameter values. When registering mobile workflow devices, an administrator can choose one of four device settings. The customer then populates the settings with whatever values they like. Custom parameters can also be set programmatically through Web services.

      Note: If you are editing properties of a rule created by a developer, you cannot modify the matching rules.

   b) Choose the type of search expression:
      - Equals – the field must exactly match the text in the label.
      - Begins with – the field must begin with the text in the label.
      - Ends with – the field must end with the text in the label.
      - Contains – the text in the label must exist somewhere in the field.
      - Regular expression – search for text that matches the pattern defined by the regular expression. You can create an expression with Boolean operators, groups, or wildcards like "?" or "*". Unwired Platform uses the Boost regular expression engine. See the Boost documentation on regular expression syntax at *http://www.boost.org/doc/libs/1_40_0/libs/regex/doc/html/boost_regex/syntax.html*.

   c) Configure the text to search against, or define a regular expression in the **Value** field.

Next

Test all new and changed rules to ensure they work as designed.

Testing Configured Matching Rules

Test a new or modified matching rule to ensure it is configured correctly.

1. In the **Matching Rules** tab for a selected mobile workflow package, click **Test**.
2. Populate the fields to create a sample e-mail message against which the rule configuration is tested. Review the results:

- If a pattern for a corresponding field in the rule matches, the word `Pass` appears adjacent to the field.
- If a pattern for a corresponding field in the rule does not match, the word `Fail` appears adjacent to the field.
- Otherwise, `No Rule` appears, indicating that no rule was created for this corresponding e-mail field.

Configuring Context Variables

Configure context variables to customize how data is loaded into the Unwired Server cache. By determining the context variables you want to use, you may create a data set that is smaller, more focused, and yields better performance.

Developers define context variables for the mobile business object. The administrator can change some of the values of a selected variable, should the design-time value need to change for a production environment. Which values are configurable depends on whether the developer hard-coded a set of user credentials or used a certificate.

1. In the left navigation pane, click **Workflows** > **MyWorkFlow**.
2. In the right administration pane, click the **Context Variables** tab.
3. To change the password or a certificate:
   a) Select SupPassword and click **Modify**.
   b) Type a new production value for the password.
      The characters you enter are obfuscated upon entry.
   c) If you want to select a new certificate file, click **Browse** to upload a production version of the certificate then click **OK**.
      The read-only values of SupUser, SupCertificateSubject, SupCertificateIssuer, SupCertificateNotAfter, SupCertificateNotBefore change to reflect values of the new certificate and password you set.
4. To change values of a hard-coded set of user credentials:
   a) Select one or both of the variables: SupUser or SupPassword, and click **Modify**.
   b) Type the new value and click **OK**.
5. In the Context Variable dialog, change the value of the named variable and click **OK**.

Assigning and Unassigning Device Users

Assign mobile workflow packages to make them available to a device user. Unassign them when a package is no longer required.

1. In the left navigation pane of Sybase Contorl Center, click **Workflows** > **MyWorkFlow**.
2. In the right administration pane, click the **Devices** tab.
3. Locate the device to assign a mobile workflow package to, then:

a) Click **Assign Workflow**.
b) List the activation users to assign the mobile workflow package to.

   By default, no users are listed in this window. Search for users by selecting the user property you want to search on, then selecting the string to match against. Click **Go** to display the users.
c) Click **OK**.

4. To unassign a mobile workflow package, select the Activation User Name and click **Unassign Workflow**.

Checking Mobile Workflow Users and Queues

Check mobile workflow application users and review pending activities for a mobile workflow application.

1. In the left navigation pane, expand the **Workflows** folder and select the mobile workflow you want to administer.
2. To check mobile workflow users:
   a) In the right administration pane, select the **Devices** tab.
   b) Review data about mobile workflow device users:
      - Activation User Name – the name of the user that activates the device.
      - Device ID – the unique identifier attached to the device hardware of a registered device.
      - Errors – the total number of errors on the device.
      - Transform Items – the total number of items in the transform queue. The transform queue contains items that Unwired Server has transformed from e-mail messages into mobile workflow messages to be sent to clients.
      - Response Items – the total number of items in the response queue. The response queue contains mobile workflow messages that are sent from the device to Unwired Server.
      - Device Number – the unique identifier for a registered mobile device that is generated and maintained by Unwired Server.
      - Transform Queue Status – the current status of the transform queue: active, awaiting credentials, or awaiting retry.
      - Response Queue Status – the current status of the response queue: active, awaiting credentials, or awaiting retry.
3. To view pending activities for a mobile workflow:
   a) Select the **Queue Items** tab.
   b) Review data about pending mobile workflow activities:
      - Activation User Name – the name of the user that activates the device.
      - Queue Type – the type of mobile workflow queue: response or transform.

- Device ID – the unique identifier attached to the device hardware of a registered device.
- Device Number – the unique identifier for a registered mobile device that is generated and maintained by Unwired Server.
- Queue ID – the unique identifier of the queued item.
- State – the status of the mobile workflow queue: active, awaiting credentials, or awaiting retry.
- Creation Date – the date the queue item was created.
- Retry Date – the date that the processing of the queue item is scheduled to be retried (if applicable).

4. To manage the mobile workflow queue in the event of non-recoverable errors:
   a) Select the **Queue Items** tab.
   b) Identify a workflow queue item that requires you to unblock it or delete it.

      Errors affecting workflow queue items are either recoverable (where a retry is applicable) or unrecoverable/unknown (where no automatic retry occurs, or there is very long retry interval).

      To recover from a long retry interval, an administrator can unblock a queue currently in retry state, so the next work schedule can pick up the blocked item immediately, instead of waiting for the retry timeout.
   c) Select one of more of the queue items from the same queue type (the queue type for all the selected items must be either Transform or Retry).
   d) Select one of the following actions:
      - **Delete** – deletes the selected workflow queue item(s).
      - **Unblock** – unblocks the selected workflow queue item(s) that are currently in a retry state.
   e) Click **OK** to confirm the action.

Deploying a Mobile Workflow Package

Use the Deploy wizard to make mobile workflow packages available on the Unwired Server.

1. In the left navigation pane, click **Workflows**.
2. From the **General** tab, click **Deploy**.
3. Click **Browse** to locate the Mobile Workflow package.
4. Select the file to upload and click **Open**.
5. Select the deployment mode:
   - New – generates and deploys the mobile workflow package and its files for the first time.

If the uploaded file does not contain a Mobile Workflow, or a Mobile Workflow with the same name and version is already deployed to Unwired Server, you see an error message.

- Update – updates the existing mobile workflow package with the newly generated mobile workflow package and its files before deploying. The previous version of the package remains on the server
- Replace – replaces any pre-existing mobile workflow packages while preserving any user assignments.

The package is added to the list of deployed packages, which are sorted by Display Name.

Next

Configure the package if you want the deployed package to have a different set of properties in a production environment.

Configuring Sybase Control Center

Configure Sybase Control Center to monitor and manage your resources.

Security

The Sybase Control Center security model delegates user authentication to the operating system or to your LDAP server.

When Sybase Control Center authenticates through the operating system, it uses the operating system of the Sybase Control Center server machine (not the client). Sybase Control Center requires each authenticated login account to have a predefined role. When a login is authenticated, roles for the login are retrieved by the security module and are mapped to Sybase Control Center predefined roles. Authorization is resolved through the mappings between the security module native roles and Sybase Control Center roles. You can enable mappings by creating a "sybase" group in your operating system or LDAP server and adding all Sybase Control Center users, or by modifying the Sybase Control Center `roles-map.xml` file to configure the mapping of native roles to Sybase Control Center roles. The security model authenticates the logins and authorizes access to managed resources.

Sybase Control Center provides a set of three predefined login modules for authentication. All login modules are defined in the `<install_location>/SCC-3_0/conf/csi.properties` file. The syntax is defined by the Sybase Common Security Infrastructure (CSI) framework. You can configure the different login modules to customize security strength. The three login modules are:

- Simple Login – defines a user name, password, and a list of roles. The default user name is “sccadmin” with a blank password and a native role of “sccAdminRole”. You can create additional accounts by adding simple login modules to `csi.properties`. However,

Sybase does not recommend the use of simple login modules for authentication in production environments.

- NT Proxy Login – delegates authentication to the underlying Windows operating system. When you log in to Sybase Control Center through an NT Proxy Login module, enter your user name in the format *username@nt-domain-name*. For example, user@sybase. Windows authentication is enabled by default, but it requires some configuration.
- UNIX Proxy Login – delegates authentication to the underlying UNIX or Linux operating system using Pluggable Authentication Modules (PAM). UNIX authentication is enabled by default, but it requires some configuration.

In addition, you can add an LDAP login module that delegates authentication to an LDAP server you specify.

Configuring Security Providers

Configure Sybase Control Center security providers to authenticate user logins through an LDAP server, the operating system, or both.

Sybase strongly recommends that you use a common authentication provider for all Sybase products, including Sybase Control Center. A common authentication provider ensures that single sign-on works for users of Sybase Control Center and its managed servers.

When authentication is accomplished through LDAP or an operating system, the authenticated user is assigned roles matching the groups that the user is a member of in the LDAP server or the operating system. Unless the groups have default mappings in the Sybase Control Center roles-map.xml file, you must then map those roles to Sybase Control Center roles in the roles-map.xml file.

Sybase Control Center can be configured to authenticate through any LDAP server that supports the inetOrgPerson (RFC 2798) schema.

Configuring Authentication for Windows

Authentication through the Windows operating system is enabled by default, but it requires some configuration. First, set Sybase Control Center to create an account when a Windows user logs in to Sybase Control Center.

This task is optional. However, if you choose not to create Sybase Control Center accounts automatically as described here, you must enter them manually. Sybase Control Center needs the accounts for purposes of setting authorization (user privileges).

1. Log in to Sybase Control Center using an account with administrative privileges (sccAdminRole).
2. Select **Application > Administration > Security.**
3. Check the box labeled **Automatically add SCC login records for authenticated logins**.
4. Click **OK** to close the Security dialog.

Next

There are two next steps:

- If you opted not to automatically create Sybase Control Center login accounts, follow the steps in Adding a Login to the System to enter each account into Sybase Control Center manually.
- Whether you add accounts automatically or manually, you must also grant privileges to the login accounts. You can grant privileges by assigning Sybase Control Center roles directly to the login accounts, or by assigning the login accounts to groups and mapping Sybase Control Center roles to the groups. The group approach is generally more efficient.

Configuring an LDAP Authentication Module

Configure an LDAP authentication module for Sybase Control Center by editing the security properties file to point to the correct LDAP server.

1. Open the `<SCC-install-dir>\conf\csi.properties` file.

   The location of this file indicates that the provider you are editing is intended for use with Sybase Control Center only.
2. Add to the properties file a module similar to the sample below, specifying the LDAP server that will provide user authentication.

   The sample in this step shows the properties used for an OpenDS LDAP server. See the example at the end for values that work for ActiveDirectory. Configuration properties you can use in the LDAP module are described in *LDAP Configuration Properties* on page 136.

   Each line of the LDAP server module of the properties file must begin with "CSI.loginModule." followed by a module number. (The module number in this sample is 7.) The module number you assign must be unique in the properties file, and you must use the same module number in every line of the module.

   ```
   CSI.loginModule.
   7.options.AuthenticationSearchBase=ou=users,dc=example,dc=com
   CSI.loginModule.7.options.BindDN=cn=Directory Manager
   CSI.loginModule.7.options.BindPassword=secret
   CSI.loginModule.7.options.DefaultSearchBase=dc=example,dc=com
   CSI.loginModule.7.options.ProviderURL=ldap://localhost:10389
   CSI.loginModule.
   7.options.RoleSearchBase=ou=groups,dc=example,dc=com
   CSI.loginModule.7.options.ServerType=openldap
   CSI.loginModule.7.options.moduleName=LDAP Login Module
   CSI.loginModule.7.controlFlag=sufficient
   CSI.loginModule.
   7.provider=com.sybase.ua.services.security.ldap.LDAPLoginModule
   ```

   Note: Change the values of bolded lines only.
3. Save the file.

4. If your LDAP server's SSL certificate is signed by a nonstandard certificate authority (for example, if it is a self-signed certificate), use the **keytool** utility to configure your JVM or JDK to trust the certificate. Execute a command similar to this:

```
keytool -import -keystore <sybase-dir>/shared/JRE-6_0_6/bin/
keytool/lib/security/cacerts -file
<your cert file and path> -alias ldapcert -storepass changeit
```

LDAP configuration values for ActiveDirectory

For an ActiveDirectory server, use these values for configuration properties in your LDAP login module:

```
ServerType: msad2K
DefaultSearchBase: dc=<domainname>,dc=<tld> or o=<company
name>,c=<country code>
                E.g. dc=sybase,dc=com or o=Sybase,c=us
ProviderUrl: ldaps://<hostname>:<port>
                E.g.:  ldaps://myserver:636
AuthenticationFilter: (&(userPrincipalName={uid})
(objectclass=user))
BindDN: <User with read capability for all users>
BindPassword: <Password for BindDN user>
RoleFilter: (|(objectclass=groupofnames) (objectclass=group))
controlFlag: sufficient
```

Next

Follow the steps in *Setting Up Roles and Passwords* on page 161.

Setting Up Roles and Passwords

Set the initial user roles and passwords required for Sybase Control Center to authenticate through an LDAP server.

Prerequisites

Configure an LDAP authentication module.

Task

1. Open the `<SCC-install-dir>\conf\roles-map.xml` file and add an LDAP login module.

   Insert an LDAP login module similar to this at the end of the security-modules portion of the file, just before `</security-modules>`:

```
    <module name="LDAP Login Module">
      <role-mapping modRole="sybase"
uafRole="uaAnonymous,uaPluginAdmin,sccUserRole" />
      <role-mapping modRole="administrators"
```

```
uafRole="uaAnonymous,sccAdminRole" />
    </module>
```

2. Ensure that the roles defined in the LDAP repository match the roles defined in `roles-map.xml`.
3. In the `<SCC-install-dir>\conf\csi.properties` file, set the BindPassword and ProviderURL properties with values used in your deployment.

   Sybase recommends that you encrypt sensitive values before saving them in `csi.properties`. See *Encrypting a Password* on page 162 for instructions.

Next

Follow the steps in *Mapping Sybase Control Center Roles to LDAP or OS Groups* on page 174.

Encrypting a Password

Use the **passencrypt** utility to encrypt passwords and other values that must be kept secure while stored in text files.

You can safely store an encrypted password in a properties file. Enter the password in clear text (unencrypted) when you execute **passencrypt** and when you use the password to log in.

passencrypt, which is located in the Sybase Control Center `bin` directory, uses the DES encryption algorithm.

1. Open a command window and change to the `bin` directory:

   Windows: `cd %SYBASE_UA%\bin`
   UNIX: `cd $SYBASE_UA/bin`
2. Encrypt a password:

   ```
   passencrypt -text <new_password>
   ```

   The passencrypt utility encrypts the password you enter and displays the password in encrypted form.
3. Copy the encrypted password.
4. Paste the encrypted password where needed.
5. When you have encrypted all the passwords you need, immediately close the command window—it displays passwords in clear text.

LDAP Configuration Properties

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.

- The **LDAPLoginModule** provides authentication services. Through appropriate configuration, you can enable certificate authentication in LDAPLoginModule.
- Optional. The **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 always used with every security configuration; however it is not displayed in Sybase Control Center.

  Only use LDAPAuthorizer 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 configure properties for it explicitly. It cannot share the configuration options specified for the LDAPLoginModule (if any are configured).
- Optional. The **LDAPAttributer** provides attribution services. The attribution provider can share configuration defined for the LDAPLoginModule. That is, if no explicit configuration properties are specified for LDAPAttributer, it uses the configuration information from the LDAPLoginModule, when only one LDAPLoginModule is configured. If there are more than one, then LDAPAttibuters cannot share properties, because they would not know which LoginModule to share with; in this case you must also configure propertes for each LDAPAttributer.

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). Note that Development Editions of Unwired Platform include an OpenDS LDAP server that runs on a nonstandard port of 10389. However, most LDAP servers use the standard port of 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 "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.
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. See the*Java Sun* Web site for more information.

Property	Default Value	Description
AuthenticationFilter	For most LDAP servers: `(&(uid={uid})(objectclass=person))` or For Active Directory email lookups: `(&(userPrincipalName={uid})(objectclass=user))[ActiveDirectory]` For Active Directory Windows username lookups: `(&(sAMAccountName={uid})(objectclass=user))`	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.
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 value is not specified, the LDAP DefaultSearchBase is used.

Property	Default Value	Description
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.
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. If you use an encrypted the password using the CSI encryption utility, append .e to the property name. For example: `CSI.loginModule.7.options.BindPassword.e=1-AAAAEgQQOLL+LpXJO8fO9T4SrQYRC9lRT1w5ePfdczQTDsP8iACk9mDAbm3F3p5a1wXWKK8+NdJuknc7w2nw5aGJlyG3xQ==`
RoleSearchBase	none	The search base used to retrieve lists of roles. If this value is not specified, the LDAP DefaultSearchBase is used.

Property	Default Value	Description
RoleFilter	For SunONE/iPlanet: `(&(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 or this property is not initialized, no roles are available.
RoleMemberAttributes	For Netscape Directory Server: member,uniquemember	The role's member attributes defines a comma-delimited list of attributes that roles may have that define a list of DN's of people who are in the 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 for retrieved roles that is the common name of the role. If this value is "dn" it is interpreted specially as the entire dn of the role as the role name.
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
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.
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".
DigestMD5AuthenticationFormat	DN For OpenLDAP: username	The DIGEST-MD5 bind authentication identity format.
UseUserAccountControlAttribute	For most LDAP servers: false For ActiveDirectory: true	The UserAccountControl attribute to be 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	Indicates whether authentication with this login module is sufficient to allow the user to log in, or whether the user must also be authenticated with another login module. Rarely set to anything other than "sufficient" for any login module. **Note:** controlFlag is a generic login module option rather than an LDAP configuration property.

Authorization

Defines the authorization mechanism in Sybase Control Center: the login account types and the roles.

Access to Sybase Control Center is controlled by login accounts. You grant permissiongs to a login account by assigning predefined roles which control tasks the user can perform in Sybase Control Center, such as administration and monitoring of particular types of Sybase servers. The roles can be assigned directly to login accounts or to groups; a login account inherits the roles of any group to which it belongs. Sybase Control Center classifies roles as follows:

- System roles – define how a user can interact with Sybase Control Center.
- Product roles – define how a user can interact with any managed resource in Sybase Control Center, for example a Replication Server or an Adaptive Server.

Adding a Login to the System

Use the security configuration options to create a new login account.

Prerequisites

- You must have administrative privileges (sccAdminRole) to perform this task.
- If you intend to use Windows or UNIX to authenticate users, configure the appropriate authentication module.

Task

1. From the menu bar, select **Application** > **Administration**.
2. In the Sybase Control Center Properties dialog, expand the **Security** folder.
3. Select **Logins**.
4. Click **Create Login**.
5. Enter **Login Name** and **Expiration** for the new login. Expiration is optional.
6. Click **Next**.
7. Select **Specify new user information**.
8. Specify:
   - Title
   - First Name*
   - M.I
   - Last Name*
   - Suffix
   - Email Address*
   - Phone

- Ext.
- Fax
- Mobile

*The **First Name**, **Last Name**, and **Email Address** are mandatory fields.

9. Click **Finish**.

Note: If you are using Simple Login as your predefined login module for authentication, the default login account, "sccadmin," comes with a blank password. To change or modify the password, configure the `csi.properties` file as described in the *Installation Guide*.

Next

Grant privileges to the new login account. You can grant privileges by assigning Sybase Control Center roles directly to the login accounts, or by assigning the login accounts to groups and mapping Sybase Control Center roles to the groups. The group approach is generally more efficient.

Removing a Login from the System

Use the security configuration options to delete a login account.

Prerequisites

You must have administrative privileges to perform this task.

Task

1. From the menu bar, select **Application** > **Administration**.
2. In the Sybase Control Center Properties dialog, expand the **Security** folder.
3. Select **Logins**.
4. Select the login to delete.
5. Click **Delete**.
6. Click **OK** to confirm the deletion.

Assigning a Role to a Login or a Group

Use the security configuration options to add one or more roles to a login account or to a group. Roles enable users to perform tasks such as monitoring servers or administering Sybase Control Center.

Prerequisites

You must have administrative privileges to perform this task. To assign a monitoring role for a server, first register the server.

Task

Assign the sccAdminRole to any login account that will perform administrative tasks in Sybase Control Center.

1. From the menu bar, select **Application** > **Administration**.
2. In the Sybase Control Center Properties dialog, expand the **Security** folder.
3. Click **Logins** or **Groups**.
4. In the table, select the login account or group to which you want to assign a role.
5. Click the **Roles** tab.
6. In the **Available roles for resource** list, select the role, then click **Add**. For example, to grant administrative privileges, add the SCC Service:sccAdminRole. To grant monitoring privileges, add the MonitorRole for the desired server and server type.

   If a role appears in the **Has following roles** list, this account or group has already been configured with that role.
7. Click **OK**.

Removing a Role from a Login or a Group

Use the security configuration options to remove one or more roles from a login acount or from a group.

Prerequisites

You must have administrative privileges to perform this task.

Task

1. From the menu bar, select **Application** > **Administration**.
2. In the Sybase Control Center Properties dialog, expand the **Security** folder.
3. Click **Logins** or **Groups**.
4. Select the login account or group from which you want to remove a role.
5. Click the **Roles** tab.
6. Select the role, then click **Remove**.
7. Click **OK**.

Adding a Group

Use the security configuration options to create a new group.

Prerequisites

You must have administrative privileges to perform this task.

Task

1. From the menu bar, select **Application** > **Administration**.
2. In the Sybase Control Center Properties dialog, expand the **Security** folder.
3. Select **Groups**.
4. Click **Create Group**.
5. Enter **Group Name** and **Description**.
6. Click **Finish**.

Removing a Group

Use the security configuration options to remove a group.

Prerequisites

You must have administrative privileges to perform this task.

Task

1. From the menu bar, select **Application** > **Administration**.
2. In the Sybase Control Center Properties dialog, expand the **Security** folder.
3. Select **Groups**.
4. Select the group to remove.
5. Click **Delete**.
6. Click **OK** to confirm the deletion.

Adding a Login to a Group

Use the security configuration options to add one or more logins to a group.

Prerequisites

You must have administrative privileges to perform this task.

Task

1. From the menu bar, select **Application** > **Administration**.
2. In the Sybase Control Center Properties dialog, expand the **Security** folder.
3. Click **Groups**.
4. Select the group to which you want to assign a login.
5. Click the **Membership** tab.
6. Select the login, then click **Add**.
7. Click **OK**.

Removing a Login from a Group

Use the security configuration options to remove one or more logins from a group.

Prerequisites

You must have administrative privileges to perform this task.

Task

1. From the menu bar, select **Application** > **Administration**.
2. In the Sybase Control Center Properties, expand the **Security** folder.
3. Select **Groups**.
4. Select the group from which to remove members.
5. Click the **Membership** tab.
6. Select the login, then click **Remove**.
7. Click **OK**.

Modifying a User Profile in the System

Use the security configuration options to modify the user login information.

Prerequisites

You must have administrative privileges to perform this task.

Task

1. From the menu bar, select **Application** > **Administration**.
2. In the Sybase Control Center Properties dialog, expand the **Security** folder.
3. Select **Logins**.
4. Select the login to modify.
5. Click the **User Info** tab.
6. Make changes to the user information.
7. Click **Apply**.

Mapping Sybase Control Center Roles to LDAP or OS Groups

To grant Sybase Control Center privileges to users authenticated through LDAP or the operating system, associate roles used in Sybase Control Center with groups in the operating system.

You can configure Sybase Control Center to enable users to authenticate through their local operating system or through an LDAP server. To make this type of authentication work, Sybase Control Center roles must be mapped to groups that exist in the system providing authentication (LDAP or the operating system) or in the login module. By default, Sybase

Control Center maps the 'sybase' group to Sybase Control Center roles to provide basic privileges. The table lists additional default mappings of OS groups to Sybase Control Center roles.

Login module	OS or LDAP group	Sybase Control Center roles
UNIX Proxy	root	uaAnonymous, uaAgentAdmin, uaOSAdmin
	sybase	uaAnonymous, uaPluginAdmin, sccUserRole
	user	uaAnonymous, uaUser
	guest	uaAnonymous, uaGuest
NT Proxy	Administrators	uaAnonymous, uaAgentAdmin, uaOSAdmin
	sybase	uaAnonymous, uaPluginAdmin, sccUserRole
	Users	uaAnonymous, uaUser
	Guests	uaAnonymous, uaGuest
LDAP*	sybase	uaAnonymous, uaPluginAdmin, sccUserRole

*See *Setting Up Roles and Passwords* on page 161 for instructions on adding an LDAP login module.

There are two ways to accomplish the mapping:

- (Recommended) Add a sybase group to the operating system or LDAP server Sybase Control Center is using to authenticate users, and add all users who need to access Sybase Control Center to the sybase group.
  If you are configuring authentication through LDAP, you must also perform the steps in Setting Up Roles and Passwords.
- Configure Sybase Control Center to use an existing group in LDAP or the operating system by editing the roles-map.xml file. This option is described here.

1. If Sybase Control Center is running, shut it down.
2. In a text editor, open this file:

   Windows: %SYBASE_UA%\conf\roles-map.xml

   UNIX: $SYBASE_UA/conf/roles-map.xml
3. Locate the appropriate login module (LDAP, Unix, or NT (for Windows)).
4. Copy the line that maps the sybase group and paste it into the module just above the original sybase line.
5. Change "sybase" to the name of the group in your operating system to which Sybase Control Center users belong.
   For example, if the group is SCCusers, the new line should look like this:

```
<role-mapping modRole="SCCusers"
uafRole="uaAnonymous,uaPluginAdmin,sccUserRole" />
```

6. Save the file and exit.
7. Start Sybase Control Center.

Logins, Roles, and Groups

Sybase Control Center includes predefined logins and roles.

In Sybase Control Center, a login is a database object that identifies a user who can connect to the application. A login may have roles that specify the responsibilities.

Sybase Control Center comes with a predefined login and a set of roles.

Table 15. Predefined login

Login	Description
sccadmin	Access to all the administration features in Sybase Control Center.

A role is a predefined profile that can be assigned to a login. Roles outline the access rights for a login.

Table 16. Predefined roles

Role	Description
sccUserRole	Provides non-administrative access to a Sybase Control Center user
sccAdminRole	Provides administrative privileges for managing Sybase Control Center
repMonitorRole	Provides administrative privilege to monitor replication
aseMonitorRole	Provides administrative privilege to monitor Adaptive Server

A group is made up of one or more logins with common role permissions. In Sybase Control Center you can create your own groups to suit your business requirement.

Perspectives

A perspective is a visual container for a set of views, each of which can be customized to represent different ways to look at a managed resource, such as a server, database, device, or table.

A perspective is the main workspace on the Sybase Control Center window. Only one perspective is visible at a time; however, from within that perspective, you can open and interact with multiple views.

Creating a Perspective

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

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

Removing a Perspective

Delete an existing perspective window.

1. Select the perspective tab you want to delete.
2. In the menu bar, select **Perspective > Delete**.
3. Click **OK** to confirm the deletion.

Renaming a Perspective

Change the name of your perspective for current and future sessions.

1. Select the perspective tab you want to rename.
2. From the menu bar, select **Perspective** > **Rename**.
3. Enter the new name for your perspective.
4. Click **OK**.

Resources

In Sybase Control Center, a resource represents a unique Sybase product component or subcomponent.

Sybase products comprise many components, such as servers, databases, devices, processes, and data collections. A managed resource is a product component or subcomponent that Sybase Control Center allows you to monitor and manage.

The Resource Explorer window provides you the ability to register your Sybase product component with the Sybase Control Center. You can monitor the availability and performance of each resource and based on the resulting data, adjust or reallocate the resources to meet the needs of your system environment.

Registering a Resource

Make Sybase Control Center aware of a resource (for example, a server that can be monitored) and its connection information by registering the resource.

1. In the Resource Explorer, select **Resources > Register**.
2. Specify:

Table 17. New resource type information details

Field	Description
Resource Name*	Name of the resource to register. Enter the actual name of the server, including upper and lower case letters. If the name registered in Sybase Control Center does not exactly match the server name, some monitoring functions, including the topology view, do not work.
Resource Type	Select a type: • ASE Server, Replication Only (12.5.0.0) – monitor only the RepAgent threads for an Adaptive Server that is older than version 15.0.2.0. Choose this type for an Adaptive Server that is part of a Replication environment. • Replication Agent (15.2.0.0) – monitor Replication Agent. • Replication Server (15.2.0.0) – monitor Replication Server. • ASE Server (15.0.2.0) – monitor Adaptive Server 15.0.2.0 or later. Choose this type for full Adaptive Server monitoring capabilities.
Description	A brief description to help you identify the resource

*Resource Name is a mandatory field.

Note: For information on Unwired Platform resource registration, see Basics | About Unwired Platform.

3. Click **Next**.
4. Specify the connection information for your resource:

Table 18. New resource connection information details

Field	Description
Host Name	Local host name
Port Number	Local host port number

Note: If the resource is a Replication Server, specify the **RSSD Host Name** and the **RSSD Port Number** in addition to the information mentioned in step 4. If the resource is an Adaptive Server, you have the option of specifying **Character Set** and **Language.**

5. Click **Finish**.

Unregistering a Resource

Use the resources menu to unregister the resource from Sybase Control Center.

1. In the Resource Explorer window, select a resource you want to unregister.

2. Select **Resources** > **Unregister**.
3. Click **Yes** to confirm the removal.

Adding a Resource to a Perspective

Add a managed resource to the current perspective.

1. From the Sybase Control Center toolbar, click **Launch a resource explorer**.
2. Select the resource to add to your perspective, then perform one of these actions:
   - Select **Resources** > **Add**.
   - Drag and drop resources from the Resource Explorer window onto the Perspective Resources window. Select multiple resources by pressing the Ctrl key while selecting resources.

Removing a Resource from a Perspective

Remove a managed resource from the current perspective.

1. In the Perspective Resources window, right-click the resource and click **Remove**.
2. Click **Yes** to confirm the removal.

Searching for Resources in the Resource Explorer

Search for all your managed resources or narrow your search for a particular resource.

1. In the Resource Explorer window, enter your keyword in the **Filter string** field.
2. Perform one of these tasks:
   - Select **Match Case** to search for resources with matching case.
   - Select **Exact Case** to restrict your search on the resources that exactly match the keyword.
3. Select the column to narrow your search criteria to a particular column.

Views

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

From the framework of the perspective, you use views to monitor and manage the components and subcomponents in your products, called managed resources. For example, using a view you can monitor a managed resource from a performance or availability aspect. Also, when you create and add views to the perspective, you can arrange, minimize, maximize, and generally control their display.

In Sybase Control Center, view are dynamic based on the managed resource selected. The generic views for each resource are:

- Perspective Resources
- Heat Chart

- Resource Explorer

Managing a View

Open or remove a view in the current perspective.

You can:

Task	Action
Open a View in a Perspective	In the Perspective Resources window, right-click a resource and select the view to open.
Remove a View from a Perspective	From the current perspective window, select the view to close.

Arranging View Layout for a Perspective

Use the view layout options to manage your perspective space.

Click one of these icons from the Sybase Control Center toolbar:

- **Cascade all open views**
- **Tile all open views vertically**
- **Tile all open views horizontally**

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

Sybase Control Center embedded repository stores information related to the managed resources, user preference data, operational data, and statistics.

Sybase Control Center allows you to take a snapshot of the repository prior to performing the purge operation. Sybase Control Center copies the repository to the *`install_location`*`\sybase\SCC-3_0\backup\`*`generated_directory_name`* directory. Within the generated repository directory are two files: `scc_repository.db` and `scc_respository.slg`. Sybase Control Center does not reference an archived repository and hence it is recommended that administrators should periodically move the files to another secondary storage location to prevent the install directory size from piling up.

Configuring Repository Purging

Use the Application menu to change repository purging options.

Prerequisites

You must have administrative privileges to perform this task.

Task

1. From the menu bar, select **Application** > **Administration**.
2. Select **Repository**.
3. Select **Periodically purge repository**.
4. Enter the number of days after which to purge repository data.

   Note: By default, purge is enabled. It occurs once a day and purges data older than one day. Changing the purge frequency and other options might affect Sybase Control Center performance.
5. Select purge options:
   - Purge statistics
   - Purge alert history
   - Snapshot repository before purge
6. Click **OK**.

Manage

Use Sybase Control Center to manage your servers and resources.

Managing Unwired Platform

Use Sybase Control Center for Unwired Platform to manage the key functions of a mobile environment.

Management tasks are divided into two categories.

Routine System Maintenance Tasks

Routine maintenance of Unwired Platform plays a vital role in keeping your system running smoothly and efficiently.

Checking the Unwired Server Log

View the Unwired Server log in Sybase Control Center. Aggregated log data in the console makes server information more readily accessible and actionable.

1. In the left navigation pane, expand the Servers folder, and select the server for which you want to view log data.
2. Select **Log**.
3. In the right administration pane, select the **General** tab.
4. Scroll through the log data to view server information.

Default Unwired Server Runtime Logging

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

By default, all the components of the 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>\Servers\UnwiredServer\Repository\logging-configuration.xml`). For details, see *Sybase Control Center Online Help > Configure > Configuring Unwired Platform > Servers*.

You can view these Unwired Server logs in two ways:

- From Sybase Control Center – click **Servers** > **<ServerName>** > **Log** in the left pane. The first 150 entries initially appear in the console area, so you may need to incrementally retrieve more of the log as required, by scrolling down through the log events.
- From a text editor – browse to and open one or more of the `<UnwiredPlatform_InstallDir>\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.

Viewing the Server Log

In text 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.

Searching Server Log Data

Filter server log data according to the criteria you specify.

1. In the left navigation pane, expand the **Servers** folder and select the server for which you want to review log data.
2. Select **Log**.
3. In the right administration pane, click the **General** tab.
4. Select **Show filter criteria** to display the search pane.
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: MSG, MMS, Security, MobiLink™, DataServices, or Other.

Note: Set the log level for each component in the **Setting** tab. See *Sybase Control Center for Sybase Unwired Platform > Configure > Configuring Unwired Platform > Unwired Server > Server Logs > Configuring Server Log Settings*.

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

7. Click **Retrieve**.
   The search results appear in the right log pane.
8. To begin a new query, click **Reset** in the search panel and enter new search criteria.

Retrieving the Server Log

Update the information in the log console window.

1. In the left navigation pane, expand the **Servers** folder and select a server name.
2. Select **Log**.
   The information collected in the current log appears in the console window of the **General** tab.
3. To display the latest log data collected in the log file if time has elapsed since you last opened the log, click **Retrieve**.

Deleting the Server Log

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

1. In the left navigation pane, expand the **Servers** folder and select the server you want to configure.
2. Select **Log**.
   The information collected in the current log appears in the console window of the **General** tab.
3. To delete all data from the log file and all backup log files, click **Delete**.
4. In the confirmation dialog, click **OK**.

Checking the Domain Log

View the domain log in Sybase Control Center. While domain log data is stored in the monitoring database, aggregated log data in the console makes domain information more readily accessible and actionable.

1. In the left navigation pane, expand the **Domains** folder and select the domain for which you want to view log data.
2. Select **Log**.

3. In the right administration pane, select the **General** tab.
4. Scroll through the log data to view domain information. Optionally, select a domain component to view specific log data: **Replication**, **Messaging**, **Errors**, **Data Change Notifications**, **Device Notifications**, or **Subscriptions**.

Domain Log Data Categories

The domain log view allows you to filter data according to six different categories: Replication, Messaging, Errors, Data Change Notifications, Device Notifications, and Subscriptions. Displayed data differs depending on the filter you select.

Replication and **Messaging** data display these properties:

Category	Description
MBO	During download, the name of the MBO that the client is synchronizing with.
Operation	During upload, the operation that the client is performing; either create, update, or delete.
Start Time	The date and time that the synchronization request was initiated.
Finish Time	The date and time that this part of synchronization completed.
User	The name of the user associated with the device.
Device ID	The registered mobile device involved in the domain log event.
Package	The name of the package involved in the domain log event.

Error data displays these properties:

Category	Description
MBO	During download, the name of the MBO that the client is synchronizing with.
Operation	During upload, the operation that the client is performing; either create, update, or delete.
Time	The time and date at which the error occurred.
Message	The error message.
User	The name of the user associated with the device.
Device ID	The registered mobile device involved in the domain log event.
Package	The name of the package involved in the domain log event.

Data Change Notification data displays these properties:

Category	Description
MBO	The name of the MBO involved in the data change.
Time	The time and date at which Unwired Server received the data change notification.
Package	The name of the package involved in the domain log event.

Device Notification data (for RBS packages only) displays these properties:

Category	Description
Sync Group	The group of MBOs to which the device notification applies.
Generation Time	The time and date at which Unwired Server delivered the device notification.
User	The name of the user associated with the device.
Device ID	The registered mobile device involved in the domain log event.
Package	The name of the package involved in the domain log event.

Subscription data (for MBS packages only) displays these properties:

Category	Description
Package	The name of the package involved in the domain log event.
User	The name of the user associated with the device.
Device ID	The registered mobile device involved in the domain log event.
Type	The type of device.
Start Time	The time and date at which the subscription command was initiated.
Finish Time	The time and date at which the subscription command was completed.

Viewing the Domain Log

The domain log is aggregated in a text view. Use the **All** tab to retrieve all the domain log data using supported search options. Use the vertical scroll bar to retrieve additional segments of the log file in 150 line increments. Review domain logs by category, by clicking the corresponding tab name. If you need to export the contents, use the **Export** button to output these contents to a TXT, or XML file.

You can navigate to any page by using the **First**, **Previous**, **Next**, **Last**, and **Go to** controls.

Searching Domain Log Data

Filter domain log data according to the criteria you specify.

1. In the left navigation pane, expand the **Domains** folder, and select the domain for which you want to view log data.
2. Select **Log**.
3. In the right administration pane, select the **General** tab.
4. View the package, user, and device data for one or more activities.

   To view information for one activity:

   a) From the menu bar, select the type of log data you want to view: **Replication**, **Messaging**, **Errors**, **Data Change Notifications**, **Device Notifications**, or **Subscriptions**.

   To view information for multiple activities:

   a) From the menu bar, select **All**.
   b) Click the **Log Options** tab.
   c) Select one or more types of log data to view: **Replication**, **Messaging**, **Errors**, **Data Change Notifications**, **Device Notifications**, or **Subscriptions**.
5. Specify which packages, users, and devices to search:
   - In the **Packages** tab, select packages to include in your search. If a package does not appear in the search list (for example, if it has been deleted), add it by entering the package name and clicking **Add**. Use **Refresh** used to refresh the list of packages, users, or devices. If you want to reset the search query current packages or user list, click **Reset**.
   - In the **Users** tab, select users to include in your search. If a user does not appear in the search list (for example, if he or she has been deleted), add the user by entering the user name and clicking **Add**.
   - In the **Devices** tab, select devices to include in your search. If a device does not appear in the search list (for example, if it has been deleted), add it by entering the device name and clicking **Add**. You can also locate a device by clicking **Find device** and searching according to the available criteria.
6. Above the log console, specify a **Start Date**, **Start Time**, **End Date**, and **End Time** for the search.
7. Click **Retrieve**.
8. To begin a new query, click **Reset** in the search panel and enter new search criteria.

Deleting Contents of the Domain Log

Clear domain log data from the log console window and the monitoring database.

1. In the left navigation pane, expand the **Domains** folder and select the domain you want to configure.

2. Select **Log**.
   The information collected in the current log appears in the console window of the **General** tab.
3. Click **Delete**.
4. In the **Clean Domain Log** dialog, set the time range and packages for which you want log events deleted.

Exporting Domain Log Data

Save a segment of domain log data to a location outside of the monitoring database. Export data to back up information or to perform closer analysis of the data in a spreadsheet application.

1. In the left navigation pane, expand the **Domains** folder and select the domain to configure.
2. Select **Log**.
3. Perform a search to obtain the desired log data.
4. Click **Export**.
5. Select a file type for the exported data (.TXT, or .XML) and click **Next**.
6. Click **Finish**.
7. In the file browser dialog, select a save location and enter a unique file name.
8. Click **OK**.

Checking Client Application Logs

Review data about client application operations for all devices subscribed to a package in order to track errors and identify performance issues.

1. In the left navigation pane, expand the **Packages** folder and select the package you want to manage.
2. In the right administration pane, select the **Client Log** tab.
3. Review this information to monitor client application activity:
   - Activation User Name – the name of the user that activates the device.
   - Device ID – the unique identifier attached to the device hardware of a registered device.
   - MBO – the mobile business object that the client is synchronizing with.
   - Operation – the operation that the client is performing.
   - Code – the result of server-side operations; either 200 (successful) or 500 (failed).
   - Level – the log level for the application; either FATAL, ERROR, WARN, INFO, DEBUG, or TRACE.
   - Timestamp – the date and time at which the operation took place.
   - Message – the log message associated with the operation.
4. Select a row from the table and click **Details** to see a detailed view of data for the selected client log event.

5. Click **Close** to return to the Client Log summary view.

Cleaning the Client Log

Clears client application log data from the SCC administration page.

1. In the left navigation pane, expand the **Packages** folder and select the package you want to manage.
2. In the right administration pane, select the **Client Log** tab.
3. Click **Clean**.
4. Enter a time frame to indicate which client log data you want to erase, and click **OK**. The data is removed from the Client Log tab.

Checking System Statistics

Check statistics that reflect the performance of Unwired Server components for common activities.

1. In the left navigation pane, select **Monitoring**.
2. In the right administration pane, select a tab, depending on the type of system statistics you want to view:

Category	Description
Replication	Reflects replication-based synchronization activity for monitored packages. Allows you to identify the rate at which synchronizations happen during specified time periods, which users synchronize data, and which mobile business objects are affected.
Messaging	Reports on messaging-based synchronization activity for monitored packages. Allows you to track the progress of messages from device users, calculate the efficiency of completed transactions, and view areas of strength and weakness in the application environment.
Queue	Reflects the status of various messaging queues. Provides a historical view of messaging activities that communicates the efficiency of messaging-based synchronization, as well as the demands of device users on the system.
Data Change Notification	Monitors data change notifications (DCNs) that are received by Unwired Server from the enterprise information server. Allows you to view which packages and sync groups are affected by notifications, and how quickly these are processed by the server.
Device Notification	Monitors the occurrence and frequency of notifications sent from Unwired Server to replication-based synchronization (RBS) devices. Allow you to view the packages, synchronization groups, and devices affected by RBS synchronization requests in a given time frame.

Category	Description
Package	Reflects response times for replication- and messaging-based synchronization packages. This type of monitoring uses key performance indicators to provide data on the efficiency of response by Unwired Server to synchronization requests.
User	Reflects the overall activity of device users. Highlights key totals and identifies average, minimum, and maximum values for primary user activities.
Cache	Provides a granular view of cache activity, particularly in the areas of cache performance, mobile business object (MBO) status, and cache group status.

Checking Errors

View data pertaining to device errors in order to identify and diagnose potential configuration issues.

Aside from viewing server, domain, and package logs, administrators can check and diagnose system errors by:

Checking Mobile Workflow Errors

View or clear the error history of a mobile workflow application.

1. In the left navigation pane, expand the **Workflows** folder and select the mobile workflow you want to administer.
2. In the right administration pane, click **Error Log**.
3. Review this error-related data:
   - Time – the date and time at which the error occurred.
   - Activation User Name – the name of the user that activates the device.
   - Device ID – the unique identifier attached to the device hardware of a registered device.
   - Description – the type of error and its cause.
4. To view more detail for a particular mobile workflow error, select the box adjacent to the error and click **Details**.
5. To delete an error from the error log list, select the box adjacent to the error and click **Delete**.

Checking System Status

Check the overall status of Unwired Platform by reviewing the status of its components, including clusters, servers, and users.

These tasks are important to checking the Unwired Platform system status:

Checking Cluster Status

Verify that a cluster is running.

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

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. See *Stopping and Starting a Server* and *Suspending and Resuming a Server* for more information.

Checking Mobile Workflow Users and Queues

Check mobile workflow application users and review pending activities for a mobile workflow application.

1. In the left navigation pane, expand the **Workflows** folder and select the mobile workflow you want to administer.
2. To check mobile workflow users:
   a) In the right administration pane, select the **Devices** tab.
   b) Review data about mobile workflow device users:
      - Activation User Name – the name of the user that activates the device.
      - Device ID – the unique identifier attached to the device hardware of a registered device.
      - Errors – the total number of errors on the device.
      - Transform Items – the total number of items in the transform queue. The transform queue contains items that Unwired Server has transformed from e-mail messages into mobile workflow messages to be sent to clients.
      - Response Items – the total number of items in the response queue. The response queue contains mobile workflow messages that are sent from the device to Unwired Server.
      - Device Number – the unique identifier for a registered mobile device that is generated and maintained by Unwired Server.
      - Transform Queue Status – the current status of the transform queue: active, awaiting credentials, or awaiting retry.

- Response Queue Status – the current status of the response queue: active, awaiting credentials, or awaiting retry.

3. To view pending activities for a mobile workflow:
   a) Select the **Queue Items** tab.
   b) Review data about pending mobile workflow activities:
      - Activation User Name – the name of the user that activates the device.
      - Queue Type – the type of mobile workflow queue: response or transform.
      - Device ID – the unique identifier attached to the device hardware of a registered device.
      - Device Number – the unique identifier for a registered mobile device that is generated and maintained by Unwired Server.
      - Queue ID – the unique identifier of the queued item.
      - State – the status of the mobile workflow queue: active, awaiting credentials, or awaiting retry.
      - Creation Date – the date the queue item was created.
      - Retry Date – the date that the processing of the queue item is scheduled to be retried (if applicable).
4. To manage the mobile workflow queue in the event of non-recoverable errors:
   a) Select the **Queue Items** tab.
   b) Identify a workflow queue item that requires you to unblock it or delete it.

      Errors affecting workflow queue items are either recoverable (where a retry is applicable) or unrecoverable/unknown (where no automatic retry occurs, or there is very long retry interval).

      To recover from a long retry interval, an administrator can unblock a queue currently in retry state, so the next work schedule can pick up the blocked item immediately, instead of waiting for the retry timeout.
   c) Select one of more of the queue items from the same queue type (the queue type for all the selected items must be either Transform or Retry).
   d) Select one of the following actions:
      - **Delete** – deletes the selected workflow queue item(s).
      - **Unblock** – unblocks the selected workflow queue item(s) that are currently in a retry state.
   e) Click **OK** to confirm the action.

Checking Messaging Device User Status

Check the status of a messaging device user to determine the state of the device's connection to Unwired Server.

1. In the left navigation pane, select **Device Users**.
2. In the right administration pane, select the **Devices** tab.

3. From the menu bar, select **MBS** for messaging-based synchronization.
4. View the Status column for the user in question to check the state of the device: registered, online, offline, expired, or disabled.
5. Check Pending Items to see how many messages waiting to be delivered to the user's device.

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 tranferred to new devices.

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*.

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:
   - Production edition – the edition of the software you have installed.
   - 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 *Device User License Limits* in *System Administration*.
   - Used device license count – the total number of licenses currently attached to devices. 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. See *Device User License Limits* in *System Administration*.
   - 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.
   - 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.
   - Server license type – the type of license currently used by Unwired Platform. For more information on license types, see *Platform Licenses* in *System Administration*.
   - 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.
4. Click **Close**.

Checking and Resolving DOE-C User Failures

If the General tab of a DOE-C package displays an invalid user account error for the Error State property, you must resolve the issue by reconfiguring the username and password in the Connection Pool configured for the SAP package connection.

1. In the default domain, expand the **Packages** folder and click the DOE-C package name.
2. Check the Error State property in the **General** tab.
3. Validate the username and password configured, by clicking the **Connection** tab.
4. Correct the user credentials used by editing the corresponding connection pool properties:
   a) In the navigation pane, click **Connections**.
   b) In the administration pane, click the **Connections** tab.
   c) Select the connection for the DOE-C package, then click **Properties**.
   d) Set the username and password so that it matches the user account credentials.

   Note: If you change the username or password property of a DOE-C connection, you must reopen the same dialog and click `Test Connection` after saving. Otherwise the error state of this DOE-C package cannot be cleaned up. If you do not click `Test Connection`, the username or password is correct, but the error state of the DOE-C package cannot be cleaned up.

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.

Viewing Available Relay Servers

Reviewing MBO History

View or clear the error history of a mobile business object (MBO).

1. In the left navigation pane, expand the **Packages** folder and select the package that contains the MBO you want to view.
2. Select the MBO.
3. In the right administration pane, click the **History** tab.
4. To view MBO data from a specific time period, select a **Start date** and **End date** and click **Go**.
5. Review the following data for the MBO:
   - Data refresh time – the time of this data refresh's failure.

- Failed data refresh counts – the number of failed data refreshes that occurred during the specified time period.
- Last successful data refresh – the date and time of the most recent successful data refresh of this MBO before this refresh failure.

6. To clean MBO history data, click **Clean**.
   Data is removed from the consolidated database.

Reviewing Operation History

View the error history of a mobile business object (MBO) create, delete, or update operation.

1. In the left navigation pane, expand the **Packages** folder and select the package that contains the MBO operation you want to view.
2. Expand the desired MBO and select the operation for which you want to view the error history: **create**, **delete**, or **update**.
3. In the right administration pane, click the **History** tab.
4. To view operation data from a specific time period, select a **Start date** and **End date** and click **Go**.
5. Review the following data for the operation:
   - Operation replay time – time of this operation replay's failure.
   - Number of failed operation replays– the number of failed operation replays that occurred during the specified time period.
   - Last successful operation replay – the date and time of the most recent successful replay of this operation before this operation replay failure.
6. Click **Clean** to remove operation history data.
   The lines are removed from the consolidated database.

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 no longer referenced by any active users. • Replication-based synchronization – removes subscriptions not used since the last synchronization. • Message-based synchronization – removes subscriptions if Unwired Server has not received a synchronization message since the given date.
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 no longer referenced by any active users. • Replication-based synchronization – removes subscriptions not used since the last synchronization. • Message-based synchronization – removes subscriptions if Unwired Server has not received a synchronization message since the given date.
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.

Purging Error Log History Manually

You can manually purge error-log playback history at the package level. Online error messages are deleted. You cannot purge error history from Data Orchestration Engine Connector (DOE-C) packages.

1. In the left navigation pane, expand the **Domains** folder and select a domain.
2. Navigate to the **Packages** folder, and select a package.
3. In the right pane, select the **General** tab.
4. On the **General** tab, click **Error Cleanup**.
5. In the Purge Error History dialog, enter the number of days to be purged. The error messages that are older than the number of days specified are deleted.
6. Click **OK** to purge immediately based on your selections.

Purging the Synchronization Cache Manually

You can manually purge the synchronization at the package level. Mobile business objects (MBOs) contained in a cache group using the online policy are deleted.

1. In the left navigation pane, expand the **Domains** folder and select a domain.
2. Navigate to the **Packages** folder, and select a package.
3. In the right pane, select the **Cache Group** tab.
4. On the **Cache Group** tab, click **Purge**.
5. Click **OK** to purge immediately based on your selections.

Purging Client Log Manually

You can manually purge the client log at the package level. Client log records that have already been synchronized to the device or are no longer associated with active users are deleted.

1. In the left navigation pane, expand the **Domains** folder and select a domain.
2. Navigate to the **Packages** folder, and select a package.
3. In the right pane, select the **Client Log** tab.
4. On the **Client Log** tab, click **Purge**.
5. Enter the number of days of data to preserve. The client log entries that are older than the number of days specified are deleted.
6. Click **OK** to purge immediately based on your selections.

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).

Routine Command and Control Actions

Routine command and control are any actions that do not configure an Unwired Platform entity. Command and control actions are performed routinely according to cycles across your production environment.

Deploy

Deployment is a routine administration task that manages the life cycle of a mobile business object (MBO) package on the Unwired Server. Deployment makes a package available to the runtime environment, so that it can be administered or accessed by client devices. Deployment is similar to, but not the same as, exporting and importing packages between multiple cluster environments.

Unwired Platform supports the development and subsequent deployment of:

- Package archives
- Mobile Workflow archives

Depending on the package type, the deployment steps can vary.

Deploying a Replication or Messaging Package

Use the Deploy wizard to make packages available on the Unwired Server.

Prerequisites

If your developers have created a custom filter for the mobile business object you are deploying, you must copy class files to the primary server before you deploy the package that uses those filters. If you copy the filters to a slave server by mistake, they are deleted when you deploy the package to the primary server.

To locate the name of the master, look at the server list in Sybase Control Center. If a particular server is the master server of a cluster, it will be labeled as "primary" in the left navigation pane.

Task

Deploying is the process whereby whole or part of a mobile application is loaded onto an Unwired Server as one or more deployment units. Unwired Server can then make these units accessible to users via a client application that is installed on a mobile device.

Because the deployment unit file contains package name and other information, you do not need to select a package from the list of available packages; Unwired Server creates the package automatically according to what has been defined in the deployment file.

Note: If the connection properties of an MBO or operation use credentials that have been customized manually by a developer, the back-end data sources connection properties of these MBOs and operations cannot be updated by an administrator when using the Deploy Wizard.

Instead, the adminstrator can update these properties after the MBO or operation is deployed to an Unwired Server. When the server connection is created on Unwired Server, the administrator can then change connection properties in Sybase Control Center by clicking on the Connections node in the left navigation pane.

Launching the Deploy Wizard

Launch the Deploy wizard when the packages you require have already been created, or when you want Unwired Server to create a package for you from a deployment unit file.

1. In the left navigation pane, expand the **Domains** folder.
2. Choose a domain name, then select **Packages**.
3. In the right administration pane, click the **General** tab.
4. Click **Deploy**.
   Follow the instructions in the wizard to configure a package so it can be deployed.

Configuring Deployment Properties

Set the properties for packages being deployed on Unwired Server.

Prerequisites

Ensure that the deployment unit and the optional deployment descriptor exists and that you can browse to the location of these files. Also, before you start the deploy wizard, ensure that custom filter class files have been copied to the `<UnwiredPlatform_InstallDir>\<UnwiredPlatform-XX>\Servers\UnwiredServer\lib\filters` folder.

Task

A deployment file and an optional deployment descriptor file are created by developers in Unwired Workspace. These files are typically delivered for deployment to a production version of the Unwired Server by an administrator.

Note: If the package is deployed to the primary server, it is cluster-level operation.

1. When the **Deploy** wizard loads, click **Next**.
2. Review the **Deployment File** name, or click **Browse** to navigate to the appropriate file.

   You can select either a single deployment unit (`.XML`) or an archive file (`.JAR`).

   The name of the **Package Name** appears. If this package does not already exist, the wizard displays a message that indicates the new package will be created. The name cannot exceed 64 characters or include any periods (".").
3. Select a **Synchronization Mode**.
   - Replication – for replication-based sync packages.
   - Messaging – for messaging-based sync packages.

     Packages that use an Online cache group policy must use Messaging, Replication is unavailable for these packages.
4. Select a **Deployment Mode**.

   The deployment mode you choose depends on whether or not a package of the same name has already been deployed to Unwired Server.

Mode	Description
Update	Updates the target package with updated objects. After deployment, objects in the server's package with the same name as those being deployed are updated.
Noclobber	Deploys the package only if there are no objects in the target server's package that have the same name as any of those objects being deployed.
Replace	Replaces any of the target objects with those in the package. After deployment, the servers package contains only those objects being deployed.
Verify	Does not deploy the package. Returns only errors, if any. Used to determine the results of an update deploy mode.

5. If you did not choose a deployment archive as your deployment file, you may browse and select an optional deployment **Descriptor File**:
6. Click **Next**.
7. Select a **Domain** to deploy the package to.
8. Select a **Security Configuration** for the package.
9. Click **Next**.
   The Configure Role Mapping page appears.

Deployment Archives

An archive is produced after a developer creates a package profile and executes a build on a package. This archive can only be created in the Eclipse edition of Unwired WorkSpace.

In Unwired WorkSpace, a developer executes a build process so that it creates a `.jar` archive file, which contains both a deployment unit and a corresponding descriptor file. A deployment archive can be delivered to an administrator for deployment to a production version of the Unwired Server.

Deployment Descriptors

A deployment descriptor is an XML file that captures changes to the deployment unit during deployment. Those changes are then used when a package is redeployed.

A deployment descriptor is not required to deploy the deployment unit.

A deployment descriptor is created either after a developer creates a package profile and executes a build on a package, or when the developer deploys a package to a development edition server from Unwired WorkSpace. The file contains this information:

- The deployment mode
- The target package that descriptor applies to
- The endpoint information that overrides specific endpoints defined for MBOs or operations in the deployment unit
- The domain and named security that the package applied to
- Role mappings

This information is specific to each deployment unit; therefore, you cannot apply a descriptor from another package to a deployment unit.

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.
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.

Updating Server Connection Properties

Configure the production version of server connection properties. Typically, the endpoint and role mapping a developer would use are not the same for a production system. Administrators must reset these properties accordingly.

This step allows you to change connection profiles used for development (design time) to an appropriate server-side connection. For example, your development environment might permit access to certain systems that the Unwired Server prohibits.

Note: If the connection properties of an MBO or operation use credentials that have been customized manually by a developer, the back-end data sources connection properties of these MBOs and operations cannot be updated by an administrator when using the Deploy Wizard.

Instead, the adminstrator can update these properties after the MBO or operation is deployed to an Unwired Server. When the server connection is created on Unwired Server, the administrator can then change connection properties in Sybase Control Center by clicking on the Connections node in the left navigation pane.

---

1. Review the connection properties.
2. Depending on how the properties are configured, choose an appropriate option:
   - If the properties are configured by the developer as a set of connection properties, you can edit properties as needed. To edit a property, click a field in the **Value** column and change the value as required.
   - If the properties are configured as an endpoint, choose the connection pool you want Unwired Server to use. You cannot alter any of these properties.
3. If you want the changes to apply to operations only, click the corresponding check box at the bottom of the table.
4. Click **Next**.

The connection properties are updated. The changes you make are displayed in the Summary page in the Endpoint Updates section.

Reviewing the Deployment Summary

Review the properties you have supplied before deploying the package to Unwired Server. This allows you to change errors before making the deployment units available via the package created for that purpose.

1. Review all three sections of the deployment summary. If anything is incorrect, click **Back** and correct errors.

2. To create a deployment descriptor:
   a) Click **Create deployment descriptor**.
   b) Browse to the location you want to save the file and change the default name if required.
   c) Click **OK**.

   You can use the deployment descriptor to redeploy the deployment unit without having to repeat these steps in the deployment wizard.
3. Click **Finish** to deploy the file and create the package on Unwired Server.

   Deploying to the server may take some time to complete. However, a status message appears above the **General** tab indicating the success or failure of the attempt.

Deploying a Mobile Workflow Package

Use the Deploy wizard to make mobile workflow packages available on the Unwired Server.

1. In the left navigation pane, click **Workflows**.
2. From the **General** tab, click **Deploy**.
3. Click **Browse** to locate the Mobile Workflow package.
4. Select the file to upload and click **Open**.
5. Select the deployment mode:
   - New – generates and deploys the mobile workflow package and its files for the first time.

     If the uploaded file does not contain a Mobile Workflow, or a Mobile Workflow with the same name and version is already deployed to Unwired Server, you see an error message.
   - Update – updates the existing mobile workflow package with the newly generated mobile workflow package and its files before deploying. The previous version of the package remains on the server
   - Replace – replaces any pre-existing mobile workflow packages while preserving any user assignments.

   The package is added to the list of deployed packages, which are sorted by Display Name.

Next

Configure the package if you want the deployed package to have a different set of properties in a production environment.

Importing Package Contents

Import package contents after they have been exported from another Unwired Server.

1. In the left navigation pane, expand the **Packages** folder.
2. In the right administration pane, click **Import**.

3. Enter a local file to import, or click **Browse** to navigate to the file.
4. Click **OK**.

Exporting Packages

Export a package to bundle one or more MBOs and package options in the selected package to create a new instance of a deployment archive. Typically, export is used to move the package contents onto another Unwired Server, however you can only import packages to the same domain as it was exported from.

1. In the left navigation pane, expand the **Packages** folder.
2. In the right administration pane, check the box adjacent to the name of the package and click **Export**.
   The **Export Package** dialog appears.
3. Check the appropriate export package options from the list: **Include synchronization tracing**, **Include package logging,**,**Include role mappings**, and **Include device notification templates** (for RBS packages only).

   These selections determine which package settings are retained for the new instance of the package. The current security configuration for the package is automatically applied to the exported package.
4. Click **Next**.
5. Select the file system target for the exported contents and click **OK**.

   Note: Ensure that you do not hide file type extension when you name the export package; otherwise, when the *.zip extension becomes invisible, which adversely affects the outcome of the export process.

   A status message on the General tab indicates the success or failure of the export transaction. If successful a ZIP file is created in the location you specified. You can then use this file to import the package.

Next

Deliver the file to the appropriate person or deploy the exported package on the appropriate server.

Deleting a Package

Delete a package when you want to permanently remove all elements deployed on an Unwired Server. If you do not want to permanently prohibit access by removing these files, consider disabling the package instead.

1. In the left pane, click the Unwired Server you are currently logged in to.
   This expands the list of Unwired Platform components you can manage, provided that you have the correct permissions to do so.
2. Click **Domains** > ***DomainName*** > **Packages**.

3. To delete a package from the this list, select one or more packages and click **Delete**. A confirm dialog box appears.
4. Click **OK** to confirm the deletion.

The package is removed from Unwired Server.

Managing Deployed Package Subscriptions

Manage replication, messaging, and SAP Data Orchestration Engine connector (DOE-C) package subscriptions that specify the synchronization messages mobile device users receive.

Subscription management tasks include pinging, unsubscribing, recovering, suspending, resuming, resynchronizing, and logging subscriptions. Subscription tasks vary by the package type.

These subscription management tasks apply only to the package types specified in the table below. Perform each task in the Subscriptions tab of the deployed package you are managing.

Table 19. Subscription management tasks

Subscription task	Description	Summary	Package type
Ping	Ensure that push information a user provides for a device is configured correctly. If the ping is successful, notifications and subsequent data synchronizations occur as defined by each subscription. If the ping fails, open the log and check for an incorrect host name or port number.	Select the box adjacent to the device ID, and click **Ping**.	Replication
Unsubscribe	Remove a subscription from Unwired Server.	Select the box adjacent to the device ID, and click **Unsubscribe** for replication packages, messaging packages, and DOE-C packages. For Windows Mobile, the device application must include the DatabaseClass.CleanAllData(); method for data to be unsubscribed correctly. If this method is not used, **Unsubscribe** and **Subscribe** could work unpredictably.	All

Subscription task	Description	Summary	Package type
Recover	Reestablish a relationship between the device and Unwired Server. Perform recovery under severe circumstances when a device is unable to successfully synchronize data. During subscription recovery, Unwired Server purges all enterprise data on the device. It retains the device ID and subscription information so that all data can then be resynchronized and loaded onto the device.	Check the box adjacent to the subscription ID of the device, and click **Recover**.	Messaging
Suspend/resume	Control the deactivation and reactivation of package subscriptions: • Suspend – temporarily block data synchronization for a device subscribed to a particular package. • Resume – reactivate a package subscription after it has been suspended.	Select the box adjacent to the subscription ID of the device, and click either **Suspend** or **Resume**.	Messaging DOE-C
Resynchronize	Reactivate subscriptions to a deployed package. If a DOE-C subscription does not respond to the SAP DOE quickly enough, the DOE may mark that subscription's queues as "blocked" and stop sending messages to the DOE-C. Resynchronize to resume communication from the DOE to the DOE-C subscription.	Check the box adjacent to the subscription ID of the device, and click **ReSync**.	DOE-C
Purge	Removes subscriptions that are no longer referenced by any active users.	Select the subscription, click **Purge**, and then select the criteria.	Messaging Replication

Purging a Cache Group

Physically delete data that has been logically deleted from the cache. Cached data is marked as logically deleted when certain activities occur in the client application or back end.

1. In the left navigation pane of Sybase Control Center, expand the **Packages** folder and select the package to configure.
2. In the right administration pane, select the **Cache Group** tab.
3. Click **OK**.

Purging RBS Package Subscriptions Manually

Periodically clean up replication-based synchronization subscriptions for a specific domain.

1. In the left navigation pane, expand the **Domains** folder and select a domain.
2. Navigate to the **Packages** folder, and select a replication-based subscription.
3. In the right pane, select the **Subscriptions** tab.
4. On the **Subscriptions** tab, click **Purge**.
5. In the **Purge Subscriptions** dialog, enter the number of days to be purged. Any inactive subscriptions that are older than this value are removed.
6. Click **OK**. The subscriptions that match the purging criteria are physically deleted from consolidated database. Devices for which you have purged subscriptions cannot perform any operations.

Searching for Inactive RBS Subscriptions

At any time you can search for inactive replication-based synchronization subscriptions. This is useful for previewing subscription purge candidates.

1. In the left navigation pane, expand the **Domains** folder and select a domain.
2. Navigate to the **Packages** folder, and select a replication-based subscription.
3. In the right pane, select the **Subscriptions** tab.
4. Select **Number of Inactive Days** for the search.
5. Enter the number of inactive days, then click **Go**. The subscription candidates that would be purged are retrieved.

Purging MBS Package Subscriptions Manually

Periodically clean up message-based synchronization subscriptions for a specific domain. These are subscriptions that have been inactive, and fall outside of the number of days to preserve.

1. In the left navigation pane, expand the **Domains** folder and select a domain.
2. Navigate to the **Packages** folder, and select a message-based subscription.
3. In the right pane, select the **Subscriptions** tab.

4. Click **Purge**.
5. Enter the number of inactive days to purge. Any inactive subscriptions that are older than this value are removed.
6. Click **OK**.

Searching for Inactive MBS Subscriptions

At any time you can search for message-based synchronization subscriptions. This is useful for previewing subscription purge candidates.

1. In the left navigation pane, expand the **Domains** folder and select a domain.
2. Navigate to the **Packages** folder, and select a message-based subscription.
3. In the right pane, select the **Subscriptions** tab.
4. Select the search criteria:
   - Select **Number of Inactive Days**, and enter the number of inactive days. This searches for MBS subscriptions that have not been active for the number of days you specify. For example, if you enter 90, and subscriptions that have not been active for the last 90 days are retrieved.
   - Select **Last Active Date**, and enter a date. This searches for MBS subscriptions that were active as of the date specified.
5. Click **Go** to retrieve the selection candidates that meet the criteria.

Provision

Provisioning is a routine administration task that involves all activities for managing a device, particularly with respect to access entitlements or data. Device provisioning is related to the type of application deployed to the device (for example, replication, messaging, or mobile workflow). The type of application determines what kinds of tasks are required to manage the device.

Device Users

Device users are individuals whose devices have been registered manually, through messaging-based applications, or automatically, through replication-based applications.

Device users are managed in Sybase Control Center (SCC) according to the device they use to synchronize data. The device user is a mechanism that identifies the person that controls the device. For messaging environments, the ID for this users consists of abbreviated user name. In some cases, ID of the device user can match the user name property that the user defines as part of the connection settings of the messaging client on the device.

The device users contrasts with the concept of an application user. An application user is the actual back-end EIS identity. The application user ID is the mechanism used to authenticate data request with the security provider you configure for Unwired Server. An application user ID is required when a person subscribes to a package, or when the user preforms replay operations back to the EIS.

Devices

Devices interact with Unwired Platform to gain access to corporate information. Registration of a device creates an account that identifies a user by the device registered. The user must authenticate with the authentication provider before any mobile business applications can be accessed from the device application.

Devices are categorized by the synchronization model used by the device application. The type of synchronization model used determines the configuration and administration actions you can perform.

Device Information

Access device information in the Devices tab of the Device Users node. View data on registered devices in order to manage and monitor device synchronization.

Select one of the following modes:

- **Unified** – lists all registered replication- and messaging-based synchronization devices. This view presents key information including the current registration status of the device, the device synchronization mode, and when the device last connected with Unwired Server.
- **RBS** – lists replication-based synchronization devices that are automatically registered when replication-based application users synchronize successfully with the server.
- **MBS** – lists messaging-based synchronization devices that are currently registered either manually or using the public API (for bulk registration). This view allows the administrator to see messaging device status information that is useful for diagnostic purposes.

View the following device information, depending on the filter you choose:

Property	Description	Filter
Device ID	The unique identifier attached to the device hardware of a registered device. In MBS mode, until the device is registered, the device ID remains empty string in MBS mode. In Unified or RBS mode, the device ID is `MBS_<number>`. However, the permanent ID is not assigned until the device connects to Unwired Server. For example, after Windows Mobile 6 simulator connects to the Unwired Server, the assigns an permanent ID as `Emulator324567336`.	All
Device Platform	The operating platform that the device uses.	All
Device Type	The type of device. For example, if the device model is a BlackBerry, the type is the form factor (for example, BlackBerry Bold).	All
Registered Date	The date of initial device registration.	All
Last Connected	The time at which the last communication took place between Unwired Server and the device.	All
RBS Status	The time at which the last communication took place between Unwired Server and the device.	Unified and MBS
MBS Status	The registration status of the device for messaging-based synchronization.	Unified and RBS
Lock Status	Locked or Unlocked. Synchronization is disabled in locked devices	All
Activation User Name	Tthe name of the user that activates the device.	MBS

Property	Description	Filter
Status	The current status of a device. The possible values are: Running, Suspended, Pending Activation, Online, Offline, and Expired. **Note:** If the device connection is through Relay Server, the connection to Unwired Server remains open for up to 6 minutes after the device has dropped its connection to Relay Server. Thus, the Status column may incorrectly show that the device is online for up to 6 minutes after the device has disconnected.	MBS
Pending Items	The pending items on the server side that needs to be sent to the device. The device must have connect to server at least once, before information in this column appears.	MBS
Activation Code Expire	The date the activation code expires on.	MBS
Last Delivery	The date and time of last item delivered from Unwired Server to the device. The device must have connect to server at least once, before information in this column appears.	MBS
Client Version	The Sybase Unwired Platform client runtime version of the client. The device must have connect to server at least once, before information in this column appears	MBS
Node	The cluster node on which the device is registered.	MBS

These columns can be used to sort the data by clicking the column title.

Advanced MBS Properties

If the device type is MBS, toggle **Advanced** to show or hide these columns:

Property	Description
Registered Date	The date the device was registered.
Last Connected	The date and time the device last connected.
RBS Status	Whether the MBS device is also used for RBS synchronization.
Lock Status	Whether the device has been locked.

These columns cannot be used to sort data.

Deleting Replication and Messaging Devices

Delete a device from Unwired Server.

When you delete a device, all corresponding package subscriptions related to that device are removed from the system, and the license used by that device is returned to the pool. Actual user data, such as personalization keys, is not deleted.

1. In the left navigation pane, select **Device Users**.
2. In the right administration pane, select the **Devices** tab.
3. From the menu bar, select **RBS** to view replication-based synchronization devices, **MBS** to view messaging-based synchronization devices, or **Unified** to view both types of devices.
4. Select the check box adjacent to the device you want to delete, and click **Delete**.
5. In the confirmation dialog, click **Yes**.

Locking and Unlocking Devices

Lock or unlock devices to control which users are allowed to synchronize data. Locking a device is an effective way to temporarily disable a specific user without making changes to the security profile configuration of the package to which he or she belongs.

1. In the left navigation pane, select **Device Users**.
2. In the right administration pane, select the **Devices** tab.
3. From the menu bar, select **RBS** to view replication-based synchronization devices, **MBS** to view messaging-based synchronization devices, or **Unified** to view both types of devices.
4. Select the check box adjacent to the device you want to manage, and:
   - If the device is currently unlocked and you want to disable synchronization, click **Lock**.
   - If the device is currently locked and you want to enable synchronization, click **Unlock**.
5. In the confirmation dialog, click **Yes**.

Registering and Reregistering Messaging Devices

Use Sybase Control Center to trigger the registration and device activation process, which allows messaging mobile business objects (MBOs) to handle messages belonging to different data sources.

Note: When using a Windows Mobile emulator or BlackBerry simulator to register a device in Sybase Control Center, the device ID changes each time you reset the emulator to factory settings and reinstall the client. Before reinstalling, you must delete the original device from Unwired Server. Then, reregister the device. Otherwise, the device log shows a `Wrong Device for Code` error when the device attempts to connect after registration. This problem occurs with Windows Mobile emulator and BlackBerry simulator devices.

1. In the left navigation pane, click the **Device Users** node.
2. In the right administration pane, click the **Devices** tab.
3. Click **Register** to register a new device, or **Reregister** to update the device used of an existing device user.
4. In the **Register Device** or the **Reregister Device** dialog:
   a) For new device registrations 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 device registration. If you have not created any templates, only **Default** appears in the list.
   The template you choose supplies initial values in the subsequent device activation fields.
5. Change the default activation field values for the template you have chosen. If you are using the default template, you must provide the server name, which is empty.

   If you are using a 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 letters A – Z (uppercase or lowercase), numbers 0 – 9, or a combination of both. Default: 0.

     Note: If the device uses relay server to connect to Unwired Server, the farm ID should be the name of the Unwired Server farm configured in the relay server for messaging-based synchronization applications. If the device connects to Unwired Server directly, the farm ID should be 0.
   - **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 **Activation Code** to enter the code sent to the user in the activation e-mail. This value can contain letters A – Z (uppercase or lowercase), numbers 0 – 9, or a combination of both. Acceptable range: 1 to 10 characters.

   If the activation code is automatically generated, the code for the device can be retrieved from the **Connections** group of the **Device Properties** dialog
7. Click **OK**.

Assigning and Unassigning Workflow Packages to Device Users

Assign mobile workflow packages to messaging devices make them available to an activation user. You can also unassign mobile workflow packages at any time.

1. In the left navigation pane, click **ClusterName > Device Users > Devices** tab.
2. In the right administration pane, click the **Devices** tab, select an MBS device, then click **Workflows**.
3. Select the workflow packages you want to assign or unassign.

   The Workflow Assignment dialog shows workflows that are already assigned to the selected MBS device. Select one of the workflows, then click **Unassign workflows** to unassign, or **Assign workflows** to assign it to another workflow, in the Assign Workflows dialog.
4. In the confirmation dialog, click OK.

Cloning Messaging Devices

Create a duplicate copy of a device user's messaging 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. Cloning a device requires that an additional device license be consumed.

1. In the left navigation pane, select **Device Users**.
2. In the right administration pane, click the **Devices** tab and select **MBS**. View the messaging-based synchronization devices.
3. Check the box adjacent to the device you want to clone and click **Clone**.
4. Ensure the correct values are listed in the Clone Device dialog and click **OK**.
   The cloned device appears in the list of messaging devices.
5. Check the box adjacent to the cloned device and click **Properties**.
6. Edit the configuration settings associated with the device:

   - **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 letters A – Z (uppercase or lowercase), numbers 0 – 9, or a combination of both. Default: 0.

  Note: If the device uses relay server to connect to Unwired Server, the farm ID should be the name of the Unwired Server farm configured in the relay server for messaging-based synchronization applications. If the device connects to Unwired Server directly, the farm ID should be 0.
- **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.

7. (Optional) Select the check box adjacent to **Activation Code** to enter the code sent to the user in the activation e-mail. This value can contain letters A – Z (uppercase or lowercase), numbers 0 – 9, or a combination of both. Acceptable range: 1 to 10 characters.

   If the activation code is automatically generated, the code for the device can be retrieved from the **Connections** group of the **Device Properties** dialog

Viewing Device Users

View the users registered through a particular device.

Note: Sybase Mobile CRM for SAP and Sybase Mobile Workflow for SAP application users are not registered in Unwired Server and therefore cannot be listed.

1. In the left navigation pane, select **Device Users**.
2. In the right administration pane, select the **Devices** tab.
3. From the menu bar, select **RBS** to view replication-based synchronization devices, **MBS** to view messaging-based synchronization devices, or **Unified** to view both types of devices.
4. Select the check box adjacent to the device you want to manage, and click **Users**.
5. In the View Users dialog, review this application user information:
   - Activation User Name – the name of the user associated with the device ID.
   - Security Configuration – the configuration that the user was authenticated with upon first successful authentication.
   - Register Date – the date of initial user registration.

Upgrading Replication Devices

Upgrade a replication-based synchronization (RBS) device to a messaging-based synchronization (MBS) device. This option is available only to Windows Mobile devices, which support both RBS and MBS applications.

1. In the left navigation pane, select **Device Users**.
2. In the right administration pane, click the **Devices** tab and select **RBS**. View the replication-based synchronization devices.

3. Select the box adjacent to the device you want to manage and click **Upgrade**.
4. In the Register Device dialog, select the name of the template for MBS device registration. If you have not created any templates, only **Default** appears in the list.
   The template you choose supplies initial values in the subsequent device activation fields.
5. (Optional) Change the default activation field values for the template you have chosen. If you use the default template, you must provide the server name, which is empty by default.
6. Click **OK**.

Retrieving Device Logs

Send a request to Unwired Server to retrieve log files from a messaging-based synchronization (MBS) device.

Log file retrieval is supported only for messaging devices.

1. In the left navigation pane of Sybase Control Center, select **Device Users**.
2. In the right administration pane, click the **Devices** tab, and select **MBS** to view messaging-based synchronization devices.
3. Select a device, and click **Get Trace**.

   The status of the device must be either "online" or "offline" to retrieve the log.
4. Click **OK**.
5. When the device is online, check the device log. The default locations are `<UnwiredPlantform_InstallerDir>\Data\Messaging\ClientTrace` folder of the CDB node for cluster, or `<UnwiredPlantform_InstallerDir>\UnwiredPlatform\Servers\MessagingServer\Data\ClientTrace` folder for a single node.

   Trace files, which can be either `.log` or `.txt` file types, have the following file name structure: *yyyyMMddHHmmss_UserName_DeviceName_FileName*. For example: `20091217103050_User1_Emulator67215793_Messaging_xce.log`.

Viewing Devices Associated with Users

View the devices associated with a specific application user.

1. In the left navigation pane, select **Device Users**.
2. In the right administration pane, click the **Users** tab.
3. Select the check box adjacent to the user you want to view and click **Devices**.
4. In the View Devices dialog, review this device information:
   - Device ID – the unique identifier attached to the device hardware of a registered device.
   - Device Type – the type of device. For example, if the device model is a BlackBerry, the type is the form factor (for example, BlackBerry Bold).
   - Device Platform – the operating platform that the device uses.
   - RBS status – the registration status of the device for replication-based synchronization.

- MBS status – the registration status of the device for messaging-based synchronization.
- Lock Status – either Locked or Unlocked. Synchronization is disabled in locked devices.
- Registered Date – the date of initial device registration.
- Last Connected – the time at which the last communication took place between Unwired Server and the device.
- Pending Items – the pending items on the server side that needs to be sent to the device.
- Client Version – the Sybase Unwired Platform client runtime version of the client.

Viewing Device 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 tranferred to new devices.

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*.

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:

   - Production edition – the edition of the software you have installed.
   - 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 *Device User License Limits* in *System Administration*.
   - Used device license count – the total number of licenses currently attached to devices. 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. See *Device User License Limits* in *System Administration*.
   - 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.
   - 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.
   - Server license type – the type of license currently used by Unwired Platform. For more information on license types, see *Platform Licenses* in *System Administration*.
   - 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.

4. Click **Close**.

Purging Unused Devices

Periodically clean up devices not used since a given date. This purges device associated items such as subscriptions.

1. In the left navigation pane, select the cluster, then **Device Users**.
2. In the right pane, select the **Devices** tab, and click **Purge**.
3. Specify the number of inactive days to use as the criteria for purging devices, for example, 90. Any devices that have not been used during that time period are purged.

   Use a number of days that is large enough to account for periods of time during which users may not be active users.
4. Click **OK** to purge devices, or **Cancel** to end.

Purging Unused Device Users

Periodically clean up users. User cleanup removes any user-specific data stored in the consolidated database, such as personalization keys. If you remove a security configuration, all users registered with that security configuration are also removed.

1. In the left navigation pane, select the cluster, then **Device Users**.
2. In the right pane, select the **Users** tab, and click **Purge**.
3. In the Purge Users dialog, specify:
   - **Security configuration** – select a configuration, for example, admin.
   - **Number of days since last authentication** – specify the number of days.
4. Click **OK** to purge users that meet the criteria.

Server Control

Perform server control tasks to check or change the status of a server in a cluster. Stop, ping, or start an Unwired Server.

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.

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.

Note: Restarting Unwired Server from Sybase Control Center causes the server icons in the desktop tray to disappear, even if Unwired Server is running properly. To ensure the services are running, use the Windows Services manager to see the runtime state of Unwired Server and its dependent services.

Starting and Stopping RSOE

Start and stop the RSOE process as required. However, 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 **RSOE** tab, select the RSOEs, and:
   - Click **Start**.
   - Click **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

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.

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.

Pinging a Server

Ping a server to test the availability of backend server connectivity and verify the server state (for example, started or stopped). Server ping uses an Internet Inter-ORB Protocol (IIOP) call to test if a server's IIOP 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.

Search

Searching lets you display a selected list of objects in Sybase Control Center when the list of objects is too long to scroll.

You can search various objects in Sybase Control Center for Unwired Platform; for example, device users, domain-level security, and Unwired Server log console data. The mechanism for searching is similar for all types of object:

1. Enter the search string in the text box using alphabetic or numeric characters. For alphabetic characters, the search is case-insensitive.
2. Click **Go**.

All values that meet the criteria you configured appear in the list. For example, if you type `jan` to perform a personalization-key search, and you have key values of Jane, Janet, janeb, Janus, and jAnice, all of these values are listed in the table.

Search Wildcard Reference

A wildcard character is a keyboard character such as an asterisk (*) or a question mark (?) that you can use for pattern-matching when you search for values.

Wildcards add flexibility to a value search by extending the parameters of a search string. Wildcards can help if you are uncertain of spelling, only know part of a term, or want all available spellings of a word (for example, Imperial versus American words of colour and color).

Example

For this example, there are these values that can be searched using the wildcards in the table that follows: gloss*, gloss?, glossary, Glossary_name, glossy, Glossy_name

To use a wildcard, append the search string with the appropriate wildcard combination.

Wildcard	Usage
Asterisk (*)	Substitutes for multiple characters in a string. For example, `Gloss*` locates all strings in the example because they all begin with the characters represented in the string. However, `Gloss*_name` locates all the strings that begin with "Gloss" but have the string "_name" extension, such as Glossary_name and Glossy_name.
Question mark (?)	Substitutes for a single character in a string only. For example, `Gloss?` might locate the value Glossy or Gloss1 but not Glossary.

Wildcard	Usage
Backslash and asterisk (*)	Locates any value that contains a single asterisk character "*" in the string specified. For example, `Gloss\*` locates the literal value of gloss* only.
Backslash and question mark (\?)	Locates any value that contains a single question character as in the string specified. For example, `Gloss\?` locates the literal value of gloss? only.

Monitor

Sybase Control Center allows you to monitor resource availability status, view performance statistics, and provides various log information for system administrators to debug application errors.

Monitoring Unwired Platform

Configure settings to audit the performance and availability of server and application environments.

Monitored operations include replication-based synchronization, messaging-based synchronization, messaging queue, data change notification, device notification, package, user, and cache activity. These aspects of monitoring are important to ensuring that the required data is collected.

The critical aspects of monitoring include:

1. Setting up a monitoring configuration. A monitoring configuration sets the server behavior for writing data to database, automatic purge, and data source where the monitoring data is stored.

   A default configuration is created for you, however you will likely want to customize this configuration for your environment. By default, monitoring data is flushed every 5 minutes. In development and debugging scenarios, you may need to set the flush behavior to be immediate. Set the **Number of rows** and **Batch size** properties to a low number. You can also disable flush, which results in immediately persisting changes to monitoring database. If you are setting up immediate persistence in a production environment, you may experience degraded performance. Use persistence with caution.
2. Creating a monitoring profile. A monitoring profile defines one or more domains and packages that need to be monitored.

   You can either use the **default** profile to capture monitoring data for all packages in all domains or create specific profiles as required. Otherwise, disable the **default** profile or modify it as needed.
3. Reviewing the captured data. An administrator can review monitoring data (current, historical, and performance statistics) from Sybase Control Center.

   Use the monitoring tabs to filter the data by domain, package, and time range. You can also export the data into a CSV or XML file and then use any available reporting or spreadsheet tool to analyze the data.

Monitoring Usage

Monitoring information reflects current and historical activity, and general performance during a specified time period.

Monitoring allows administrators to identify key areas of weakness or periods of high activity in the particular area they are monitoring. Access to this data helps administrators make decisions about how to better configure the application environment to achieve a higher level of performance.

The historical data is preserved in the monitor database. Performance data (KPIs for Replication, Messaging, Package Statistics, User Statistics, and Cache Statistics) for the specified time period is calculated upon request using the historical data available for that period. If monitoring data is purged for that time period, the performance data calculations will not factor in that data. It is recommended to purge monitoring data after putting in place mechanisms to export the required historical and/or performance data as needed. By default, monitoring data is automatically purged after seven days.

Also note that the processing times are calculated based on the time the request (or message) arrives on the server, and the time it took to process the request (or message) on the server. The client-side time (request origin time, and time taken to deliver to the server) are not factored into that data.

System Monitoring Overview

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:

- Replication-based synchronization
- Messaging-based synchronization
- System queue status
- Data change notifications
- Device notifications (RBS)
- Package statistics
- Device users
- 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 20. 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

Monitoring Configuration

The monitoring configuration identifies the monitoring database endpoint and determines how long data is stored in the database.

The configurable monitoring properties are:

- Auto-purge – configures an automatic purge of the monitoring database to occur on a regular basis.
- Flush threshold – determines how often monitoring data is flushed from the server memory for storage in the monitoring database.

- Flush batch size – divides flushed data into smaller segments, rather than saving all data together according to the flush threshold parameters.
- Monitor database endpoint – sets the database to be used for storage of monitoring data.

Configuring Monitoring Performance Properties

Configure auto-purge, flush threshold, and flush batch size settings to determine how long monitoring data is retained, and set a monitoring database to configure where data is stored.

Prerequisites

Depending on the expected level of monitoring activity, ensure that the monitoring database is adequately prepared to store monitoring data.

Task

1. In the left navigation pane of Sybase Control Center, select **Monitoring**.
2. In the right administration pane, select the **General** tab.
3. Click **Configuration**.
4. Configure auto purge settings.

   Auto purge clears obsolete data from the monitoring database once it reaches the specified threshold.

   a) Select **Enable auto purge configuration** to activate auto purge functionality.
   b) Enter the length of time (in days) to retain monitoring data before it is purged.
5. Configure flush threshold settings.

   The flush threshold indicates how often data is flushed from memory to the database. This allows you to specify the size of the data saved in memory before it is cleared. Alternately, if you do not enable a flush threshold, data is automatically written to the monitoring database as it is captured.

   a) Select **Enable flush threshold configuration** to activate flush threshold functionality.
   b) Select one of:
      - **Number of rows** – monitoring data that surpasses the specified number of rows is flushed from memory. Enter the desired number of rows adjacent to **Rows**. The default is 100.
      - **Time interval** – monitoring data older than the specified time interval is flushed from memory. Enter the desired duration adjacent to **Minutes**. The default is 5.
      - **Either rows or time interval** – monitoring data is flushed from memory according to whichever value is reached first: either the specified number of rows or the specified time interval. Enter the desired rows and duration adjacent to **Rows** and **Minutes**, respectively.
6. If you enabled a flush threshold, enter a **Flush batch row size** by specifying the size of each batch of data sent to the monitoring database. The row size must be a positive integer.

   The batch size divides flushed data into smaller segments, rather than saving all data together according to the flush threshold parameters. For example, if you set the flush

threshold to 100 rows and the flush batch row size to 50, once 100 rows are collected in the monitoring console, the save process executes twice; data is flushed into the database in two batches of 50 rows. If the flush threshold is not enabled, the flush batch row size is implicitly 1.

Note: By default, the monitoring database flushes data every 5 minutes. Alternatively, you can flush data immediately by removing or decreasing the default values, but doing so impacts performance and prevents you from using captured data.

7. Optional. To change the data source, select an available database from the **Monitor database endpoint** drop down list.

   Available databases are those with a JDBC server connection type (either ASE or SQL Anywhere) created in the default domain. To create a new monitor database, a platform administrator must set up a database by running the appropriate configuration scripts and creating a server connection for the database in the default domain. The database server connection then appears as an option in the Monitor Database Endpoint drop down list. For details on setting up the database, see *System Administration > Component Deployment > Deploying the Monitoring Database*.

8. Click **OK**.

Monitoring Profiles

Monitoring profiles specify a monitoring schedule for a particular group of packages. These profiles let administrators collect granular data on which to base domain maintenance and configuration decisions.

A default monitoring profile is automatically created in disabled state on Unwired Server. Administrators can enable or remove the default profile, and enable one or more new monitoring profiles as required.

The same monitoring schedule can be applied to packages across different domains; similarly, you can select individual packages for a monitoring profile.

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.

Creating and Enabling a Monitoring Profile

Specify a monitoring schedule for a group of packages.

Prerequisites

Depending on the expected level of monitoring activity, ensure that the monitoring database is adequately prepared to store monitoring data.

Task

1. In the left navigation pane, select **Monitoring**.
2. In the right administration pane, select the **General** tab.
3. Click **New** to create a monitoring profile.
4. Enter a name for the new profile.
5. Select the **Domains and Packages** tab and choose packages to be monitored according to these options:
   - Monitor all domains and packages – select **All Domains and Packages**.
   - Monitor all packages from one or more domains – select a domain, then click **Select All Packages**. Perform this step for each domain you want to monitor.
   - Monitor specific packages from one or more domains – select a domain, then select the particular packages you want to monitor from that domain. Perform this step for each domain you want to monitor.
6. Select **View my selections** to view the packages you selected for the monitoring profile. Unselect this option to return to the package selection table.
7. Select **Enable after creation** to enable monitoring for the selected packages immediately after you create the profile. By default, this option is selected. Unselect this option to enable the monitoring profile later.
8. On the **Schedule** tab, select a schedule to specify when monitoring takes place:
   - **Always On** – this schedule requires no settings. Package activity is continually monitored.
   - **Run Once** – specify a length of time during which monitoring occurs, in either minutes or hours. Package activity is monitored for the duration specified for one time only.
   - **Custom** – specify start and end dates, start and end times, and days of the week. Package activity is monitored according to the time frame specified. See *Setting a Custom Monitoring Schedule*.
9. Click **OK**.
   A status message appears in the administration pane indicating the success or failure of profile creation. If successful, the profile appears in the monitoring profiles table.
10. To enable a profile that you did not enable during creation, select the monitoring profile and click **Enable**.

Setting a Custom Monitoring Schedule

Customize the monitoring schedule for packages within a monitoring profile. Setting a custom schedule is the most flexible option; monitoring information is provided according to the time frame you specify.

Prerequisites

Begin creating a monitoring profile in the New Monitor Profile dialog.

Task

1. In the New Monitor Profile dialog, select the **Schedule** tab.
2. Select **Custom** as the monitoring schedule criteria.
3. To set a range to control which days the custom schedule runs, configure a start date and time, end date and time, or day of week (if applicable).

   - Select **Start Date** to set a date for when monitoring of package activity begins. To be more specific, you can also enter a **Start Time**. In this case, monitoring cannot begin until a given time on a given day has been reached.
   - Select **End Date** to set a date that ends the monitoring of package activity. To be more specific, you can also enter an **End Time**.
   - Select the days of the week that package monitoring runs. This means that for the days you select, the schedule runs every week on the day or days you specify.

   If you do not indicate a time frame, Unwired Server uses the default custom schedule, which is equivalent to Always On monitoring.
4. Click **OK**.

Monitoring Data

Monitoring data is aggregated in the Monitoring node of Unwired Server and organized by activity, including replication-based synchronization, messaging-based synchronization, messaging queue, data change notifications, device notifications, packages, users, and cache. The data for each activity is further broken down into current, historical, and performance-related information. View data for each monitored activity to track the performance and health of the system.

You can selectively view data accrued during a specific time period to see a snapshot of system performance during specific periods. The export function allows you to save data to a file outside of Sybase Control Center for reference or logging purposes.

Reviewing System Monitoring Data

Review data for monitored activities. The monitoring data is retrieved according to the specified time range. Key Performance Indicators (KPIs) are also calculated for the specified time range.

1. In the left navigation pane, select **Monitoring**.
2. In the right administration pane, select one of the following tabs according to the type of monitoring data you want to view:

   - **Replication**
   - **Messaging**
   - **Queue**
   - **Data Change Notifications**

- **Device Notifications**
- **Package Statistics**
- **User Statistics**
- **Cache Statistics**

Purging Monitoring Data

Clear old data from the monitoring database.

Using the Purge function in Sybase Control Center allows you to perform an ad hoc purge of monitoring data. In order to configure a regular automatic purge, see *Sybase Control Center for Unwired Platform > Monitor > Monitoring Unwired Platform > Monitoring Configuration > Configuring Monitoring Data Retention Settings*.

1. In the left navigation pane, select **Monitoring**.
2. In the right administration pane, select the **General** tab.
3. Click **Purge**.
4. Indicate the time period for which you want to delete data by specifying a **Start Date**, **Start Time**, **End Date**, and **End Time**.

   All monitoring data collected from the start time and date to the end time and date is deleted from the database. If you do not specify a start date, all data acquired prior to the end date and time is purged. Similarly, if you do not specify an end date, all data collected from the start date until the present time is purged. To save the data to file before purging it, see *Exporting Monitoring Data*.
5. Click **OK**.

A status message appears in the right administration pane indicating that the data purge was successfully completed.

Exporting Monitoring Data

Save a segment of monitoring data to a location outside of the monitoring database. Export data to back up information, particularly before purging it from the database, or to perform closer analysis of the data in a spreadsheet application.

This option is especially useful when you need to share monitoring data with other administrators and tenants. Since this task can be time-consuming, depending upon the size of the data being exported, Sybase recommends that you export the data in segments or perform the export at a time when Sybase Control Center is not in use.

1. In the left navigation pane, select **Monitoring**.
2. In the right administration pane, select the tab corresponding to the monitoring data you want to view.
3. Perform a search using the appropriate criteria to obtain the desired monitoring data.
4. Click **Export**.

5. Select a file type for the exported data (CSV or XML), and click **Next**.
6. Click **Finish**.
7. In the file browser dialog, select a save location and enter a unique file name.
8. Click **OK**.
   All monitoring data retrieved by the search is saved to the file you specify in step 7.

Searching Monitoring Data

Filter monitoring data according to a specified date and time range.

Filter options vary depending upon the type of monitoring data you search.

1. In the left navigation pane, select **Monitoring**.
2. In the right administration pane, select the tab corresponding to the monitoring data you want to view.
3. In the Search pane, indicate the time period for which you want to view data by specifying a date range to search within (that is, Start Date, Start Time, End Date, and End Time) .

   Note: You do not need to specify a time period if you are performing a search on Current date.
4. Filter the system components to include in the search.
   a) Select **Show current filter**.
   b) Specify the components to include in your search.

   If a domain or package does not appear in the search list (for example, if it has been deleted), enterthe name and click **Add**.
5. From the **Sort By** drop-down list, select a category by which to sort search results.

   Note: This field is disabled for some categories.
6. To view the domains and packages you included in the search, select **View My Selections**.
7. Click **Retrieve**.

Monitoring data for the time period specified appears in the administration console.

Monitoring Data Categories

Monitoring data is organized according to object type, allowing administrators to perform focused data analysis on specific activities and Unwired Platform components. Current, historical, and performance-based statistics facilitate user support, troubleshooting, and performance tracking for individual application environments.

The replication and messaging categories are the primary sources of data relating to application environment performance. The remaining tabs present detailed monitoring data that focuses on various aspects of replication-based applications, messaging-based applications, or both.

Replication Statistics

Replication statistics reflect replication-based synchronization (RBS) activity for monitored packages. Current statistics monitor the progress of real-time synchronizations, while historical statistics present data from completed synchronizations on a per-package basis. Performance monitoring uses key performance indicators to produce data about synchronization efficiency.

Through statistics that report on the duration and scope of synchronizations, as well as any errors experienced during synchronization, replication monitoring allows you to identify the rate at which synchronizations happen during specified time periods, which users synchronize data, and which mobile business objects are affected.

Current Replication Statistics

Current statistics for replication-based synchronization (RBS) provide real-time information about in-progress synchronizations.

Unwired Server monitors RBS requests using these statistical categories:

Category	Description
Package	The package name.
Phase	The current synchronization activity: upload or download. During the upload phase, a client initiates operation replays to execute mobile business object (MBO) operations on the back-end system. During the download phase, a client synchronizes with Unwired Server to receive the latest changes to an MBO from the back-end system.
Entity	During the download phase, the name of the MBO with which the client is synchronizing. During the upload phase, the name of the operation that the client is performing.
Synchronization Start Time	The date and time that the synchronization request was initiated.
Domain	The domain to which the package involved in synchronization belongs.
Device ID	The ID number of the mobile device participating in the synchronization.
User	The name of the user associated with the device ID.

Replication History Statistics

Historical data for replication-based synchronization consists of past synchronization details for monitored packages.

The summary view provides general information, whereas the detail view presents a more specific view of all request events during each synchronization; each row of data corresponds to a synchronization request from the client in the time frame you define:

- Click either **Details** to see more granular information on each synchronization request, or select the **Detail** option to see all synchronization request details. Detail view allows you to look at the individual messages that make up the summary view.
- Select **Summary** to see aggregated details by by domain, package, and user about past synchronization events for the defined time frame.

Table 21. Detail view information

Synchronization element	Description
Package	The package name.
Device ID	The ID number of the mobile device that participated in the synchronization request.
User	The user associated with the device ID.
Phase	The sync activity that occurred during this part of synchronization: upload or download. During the upload phase, a client initiates operation replays to change an MBO. During the download phase, a client synchronizes with Unwired Server to receive the latest changes to an MBO.
Entity	During download, the name of the MBO that the client is synchronizing with. During upload, the operation that the client is performing: create, update, or delete.
Total Rows Sent	The total number of rows sent during package synchronization. This data type is not supported at the MBO level.
Bytes Transferred	The amount of data transferred during the synchronization request.
Start Time	The date and time that the synchronization request was initiated.
Finish Time	The date and time that this part of synchronization completed.
Error	The incidence of errors during this request: true or false.
Domain	The domain to which the package involved in synchronization belongs.

Table 22. Summary view information

Category	Description
User	The name of the user associated with the device ID.
Package	The package name.
Total Rows Sent	The total number of rows sent during package synchronization.
Total Operation Replays	The total number of operation replays performed by clients during synchronization.
Total Bytes Sent	The total amount of data (in bytes) downloaded by clients from Unwired Server during synchronization.
Total Bytes Received	The total amount of data (in bytes) uploaded to Unwired Server by clients during synchronization.
Start Time	The date and time that the synchronization request was initiated.
Total Synchronization Time	The amount of time taken to complete the synchronization.
Total Errors	The total number of errors that occurred for the package during synchronization.
Domain	The domain to which the package involved in synchronization belongs.

Replication Performance Statistics

Replication performance statistics consist of key performance indicators (KPIs) that reflect the overall functioning of the application environment.

Performance monitoring highlights key totals and identifies average, minimum, and maximum values for primary activities. These calculations are dynamic, and are based on the data currently available in monitoring database for the specified time period.

All values in this table (totals, averages, maximums, minimums) apply to the specific time period you indicate:

KPI	Description
Total Distinct Package Synchronization	The total number of packages subject to synchronization.
Total Distinct Users	The total number of users who initiated synchronization requests. This value comprises only individual users, and does not count multiple synchronizations requested by the same user.

KPI	Description
Average/Minimum/Maximum Sync Time	The average, minimum, or maximum amount of time Unwired Server took to finish a complete synchronization.
Time at Minimum/Maximum Sync Time	The time of day at which the shortest or longest synchronization completed.
Package with Minimum/Maximum Synchronization Time	The name of the package and associated MBO with the shortest or longest synchronization time.
Average/Minimum/Maximum MBO Rows Per Synchronization	The average, minimum, or maximum number of MBO rows of data that are downloaded when synchronization completes.
Average/Minimum/Maximum Operation Replays per Sync (records received)	The average, least, or greatest number of operation replays per synchronization received by Unwired Server from a client.
Total Bytes Sent	The total number of bytes downloaded by clients from Unwired Server.
Total Bytes Received	The total number of bytes uploaded from clients to Unwired Server.
Total Operation Replays	The total number of operation replays performed on the EIS.
Total Errors	The total number of errors that took place across all synchronizations.
Average/Minimum/Maximum Concurrent Users	The average, least, or greatest number of users involved in concurrent synchronizations.
Time at Minimum/Maximum Concurrent Users	The time at which the least or greatest number of users were involved in concurrent synchronizations.

Messaging Statistics

Messaging statistics report on messaging-based synchronization (MBS) activity for monitored packages.

- Current monitoring data tracks the progress of messages from device users presently performing operation replays or synchronizing MBOs.
- Historical data reveals statistics indicating the efficiency of completed transactions.
- Performance monitoring provides an overall view of MBS activity intended to highlight areas of strength and weakness in the application environment.

Messaging historical data captures messages such as login, subscribe, import, suspend, resume and so on. The Import type message is a data payload message from server to client (outbound messages), while rest of the messages (login, subscribe, replay, suspend, resume) are sent from the client to server (inbound messages).

Current Messaging Statistics

Current statistics for messaging-based synchronization (MBS) provide real-time information about in-progress synchronizations. Because messaging synchronizations progress rapidly, there is typically little pending MBS data available at any given time.

Unwired Server monitors MBS requests using these categories:

Category	Description
Package	The package name.
Message Type	The type of message sent by the client to Unwired Server, indicating the current sync activity; for example, import, replay, subscribe, suspend, resume, and so on.
Entity	During the import process, the name of the mobile business object (MBO) with which the client is synchronizing. During replay, the operation that the client is performing. For all other message types, the cell is blank.
Start Time	The date and time that the initial message requesting synchronization was sent by the client.
Domain	The domain to which the package involved in synchronization belongs.
Device ID	The ID number of the mobile device participating in the synchronization.
User	The name of the user associated with the device ID.

Messaging History Statistics

Historical data for messaging-based synchronization consists of past synchronization details for monitored packages.

The summary view provides general information, whereas the detail view presents a more specific view of all request events during each synchronization; each row of data corresponds to a synchronization request from the client in the time frame you define:

- Click either **Details** to see more granular information on each synchronization request, or select the **Detail** option to see all synchronization request details. Detail view allows you to look at the individual messages that make up the summary view.
- Select **Summary** to see aggregated details by by domain, package, and user about past synchronization events for the defined time frame.

Table 23. Detail view information

Data type	Description
Package	The package name.
Device ID	The ID number of the mobile device that participated in the synchronization request.
User	The name of the user associated with the device ID.
Message Type	The type of message sent by the client to Unwired Server, indicating the sync activity; for example, import, replay, subscribe, suspend, resume, and so on.
Entity	During the import process, the name of the mobile business object (MBO) that the client is synchronizing with. During replay, the operation that the client is performing. For all other message types, the cell is blank.
Payload Size	The size of the message (in bytes).
Start Time	The date and time that the message for this sync request is received.
Finish Time	The date and time that the message for this sync request is processed.
Processing Time	The total amount of time between the start time and the finish time.
Error	The incidence of errors during this request; either true or false.
Domain	The domain to which the package involved in synchronization belongs.

Table 24. Summary view information

Category	Description
User	The name of the user associated with the device ID
Package	The package name
Total Messages Sent	The total number of messages sent by Unwired Server to clients during synchronization
Total Messages Received	The total number of messages received by Unwired Server from clients during synchronization
Total Payload Size Sent	The total amount of data (in bytes) downloaded by clients from Unwired Server during synchronization

Category	Description
Total Payload Size Received	The total amount of data (in bytes) uploaded to Unwired Server by clients during synchronization
Total Operation Replays	The total number of operation replays performed by clients during synchronization
Last Time In	The date and time that the last inbound request was received
Last Time Out	The date and time that the last outbound response was sent
Subscription Commands Count	The total number of subscription commands sent during synchronization; for example, subscribe, recover, suspend, and so on
Total Errors	The number of errors that occurred for the package during synchronization
Domain	The domain to which the package involved in synchronization belongs

Messaging Performance Statistics

Messaging performance statistics consist of key performance indicators (KPIs) that reflect the overall functioning of the application environment.

Performance monitoring highlights key totals and identifies average, minimum, and maximum values for primary activities. These calculations are dynamic, and are based on the data currently available in monitoring database for the specified time period.

All values in this table (totals, averages, maximums, minimums) apply to the specific time period you indicate:

KPI	Description
Total Messages	The total number of messages sent between the server and clients during synchronization.
Total Distinct Devices	The total number of devices involved in synchronization. This total includes the same user multiple times if he or she has multiple devices. The value comprises individual devices, and does not count multiple synchronizations requested by the same device.
Total Distinct Users	The total number of users who initiated synchronization requests. This value comprises individual users, and does not count multiple synchronizations requested by the same user if he or she uses multiple devices.
Average/Minimum/Maximum Concurrent Users	The average, minimum, or maximum number of users involved in simultaneous synchronizations.

KPI	Description
Time at Minimum/Maximum Concurrent Users	The time at which the greatest or least number of users were involved in concurrent synchronizations.
Average/Minimum/Maximum Processing Time	The average, minimum, or maximum amount of time Unwired Server took to respond to a sync request message.
Time at Minimum/Maximum Message Processing Time	The time of day at which the shortest or longest message processing event completed.
MBO for Maximum/Minimum Message Processing Time	The name of the package and associated mobile business object (MBO) with the shortest or longest message processing time.
Average/Minimum/Maximum Message Size	The average, smallest, or largest message sent during synchronization.
Total Inbound Messages	The total number of messages sent from clients to Unwired Server.
Total Outbound Messages	The total number of messages sent from Unwired Server to clients.
Total Operation Replays	The total number of operation replays performed by clients on MBOs.
Total Errors	The total number of errors that took place across all synchronizations.
Average/Minimum/Maximum Concurrent Users	The average, least, or greatest number of users involved in concurrent synchronizations.

Messaging Queue Statistics

Messaging queue statistics reflect the status of various messaging queues. The data does not reveal any application-specific information, but provides a historical view of messaging activities that communicates the efficiency of messaging-based synchronization, as well as the demands of device client users on the system.

Based on this data, administrators can calculate the appropriate inbound and outbound message queue counts for the system (configurable in the Server Configuration node of Sybase Control Center). See *Sybase Control Center online help > Configure > Configuring Unwired Platform > Unwired Server > Server Properties > Configuring System Performance Properties*.

Messaging Queue Status

Messaging queue status data provides historical information about the processing of messaging-based synchronization requests by Unwired Server. The data indicates areas of high load and times of greatest activity. This data can help administrators decide how to handle queue congestion and other performance issues.

These key indicators monitor messaging queue status:

Statistic	Description
Name	The name of the messaging queue.
Current Queued Items	The total number of pending messages waiting to be processed by Unwired Server.
Average/Minimum/Maximum Queue Depth	The average, minimum, or maximum number of queued messages. For minimum and maximum queue depth, this value is calculated from the last server restart.
Time at Minimum/Maximum Queue Depth	The time and date at which the queue reached its minimum or maximum depth.
Type	The direction of message flow: inbound or outbound.
Total Messages	The total number of messages in the queue at one point since the last server reboot.
Bytes Received	The total number of bytes processed by the queue since the last server reboot.
Last Activity Time	The time at which the most recent message was added to the queue since the last server reboot.

Data Change Notification Statistics

Data change notification (DCN) statistics monitor notifications that are received by Unwired Server from the enterprise information server. Specifically, DCN monitoring reports which packages and sync groups are affected by notifications, and how quickly these are processed by the server.

Monitoring DCN statistics allows you to troubleshoot and diagnose performance issues if, for example, the cache is not being updated quickly enough. These statistics help to identify which packages took longest to process data changes, as well as times of peak performance or strain on the system.

Data Change Notification History Statistics

Historical information for data change notifications (DCNs) consists of past notification details for monitored packages. Detailed data provides specific information on past notification activity for packages, and identifies which server data was affected.

Details about past notification events are organized into these categories:

Category	Description
Domain	The domain to which the package affected by the DCN belongs.
Package	The name of the package containing data changes.

Category	Description
MBO	The name of the MBO to which the notification applied.
Notification Time	The date and time that Unwired Server received the DCN.
Processing Time	The time that Unwired Server used to process the DCN.

Data Change Notification Performance Statistics

Data change notification (DCN) performance statistics consist of key performance indicators that reflect the efficiency of notification processing by Unwired Server.

Performance monitoring highlights key totals and identifies average, minimum, and maximum values for primary activities. These calculations are dynamic, and are based on the data currently available in monitoring database for the specified time period.

All values in this table (totals, averages, maximums, minimums) apply to the specific time period you indicate:

Key performance indicator	Description
Total Notifications	The total number of notifications sent by the enterprise information system to Unwired Server.
Average/Minimum/Maximum Processing Time	The average, minimum, or maximum amount of time Unwired Server took to process a DCN.
Time at Minimum/Maximum Message Processing Time	The time of day at which the shortest or longest DCN processing event completed.
Time of Last Notification Received	The time at which the most recent DCN was received by Unwired Server.
MBO with Minimum/Maximum Notification Processing Time	The name of the package and associated mobile business object (MBO) with the shortest or longest notification processing time.

Device Notification Statistics

Device notification statistics provide data about the occurrence and frequency of notifications sent from Unwired Server to replication-based synchronization (RBS) devices. Historical device notification monitoring reports on the packages, synchronization groups, and devices affected by RBS synchronization requests in a given time frame. Performance-related device notification data provides a general indication of the efficiency of notification processing and the total demand of synchronization requests on the system.

Device Notification History Statistics

Historical information for device notifications provides specific information on past device notifications, indicating which packages, synchronization groups, and devices were involved in synchronization requests.

Details about past device notification events fall into these categories:

Category	Description
Domain	The domain to which the package affected by the device notification belongs.
Package	The name of the package containing data changes.
Synchronization group	The synchronization group that the package belongs to.
Device ID	The ID number of the mobile device participating in the synchronization request.
Generation time	The date and time that Unwired Server generated the device notification.
User	The name of the user associated with the device ID.

Device Notification Performance Statistics

Device notification performance statistics provide a general indication of the efficiency of notification processing and the total demand of synchronization requests on the system.

Performance monitoring highlights key totals and identifies average, minimum, and maximum values for primary activities. These calculations are dynamic, and are based on the data currently available in monitoring database for the specified time period.

All values in this table (totals, averages, maximums, minimums) apply to the specific time period you indicate:

KPI	Description
Synchronization Group for Maximum Notifications	The synchronization group for which the maximum number of notifications were sent.
Package for Maximum Notifications	The package for which the greatest number of device notifications were sent.
Total Notifications	The total number of device notifications sent from Unwired Server to devices.
Total Distinct Users	The total number of users that received device notifications. This value comprises only individual users, and does not count multiple synchronizations requested by the same user.

KPI	Description
Total Distinct Devices	The total number of devices that received device notifications. This is distinct from Total Distinct Users, because a single user name can be associated with multiple devices.
Enabled Subscriptions	The total number of replication subscriptions for which notifications are generated.
Time at Last Notification	The time at which the last device notification was sent by Unwired Server.
Outstanding Subscriptions	The total number of replication subscriptions, both enabled and disabled.

Package Statistics

Package statistics reflect response times for replication-based and messaging-based synchronization packages.

This type of monitoring uses key performance indicators to provide data on the efficiency of response by Unwired Server to synchronization requests. These calculations are dynamic, and are based on the data currently available in monitoring database for the specified time period.

Replication Package Statistics

Replication package statistics consist of key performance indicators (KPIs) that reflect the overall function of the application environment at the cluster or domain level. The statistics highlight key totals and identify average, minimum, and maximum values for primary activities.

These key indicators monitor replication packages:

Note: These KPIs are not applicable at the MBO level.

- Total Bytes Received
- Total Bytes Sent
- Total Operation Replays

KPI	Description
Total Devices	The total number of devices involved in synchronization. This total includes the same user multiple times if he or she has multiple devices. The value comprises individual devices, and does not count multiple synchronizations requested by the same device.
Total Rows Sent	The total number of rows sent during package synchronization.

KPI	Description
Total Rows Received	The total number of rows received during package synchronization.
Total Errors	The total number of errors that took place across all synchronizations.
Total Bytes Received	The total number of bytes uploaded from clients to Unwired Server.
Total Bytes Sent	The total number of bytes downloaded by clients from Unwired Server.
Average/Minimum/Maximum Synchronization Time	The average, minimum, or maximum amount of time Unwired Server took to finish a complete synchronization.
Time at Minimum/Maximum Synchronization Time	The time at which the shortest or longest synchronization completed.
Total Synchronization Requests	The total number of sync requests initiated by a client.
Total Operation Replays	The total number of operation replays performed by clients on MBOs.

Messaging Package Statistics

Messaging package statistics consist of key performance indicators (KPIs) that reflect the overall function of the application environment at the cluster or domain level. The statistics highlight key totals and identify average, minimum, and maximum values for primary activities.

Note: These KPIs are not applicable at the MBO level:

- Total Subscription Commands
- Total Devices

These key indicators monitor messaging packages:

KPI	Description
Total Subscription Commands	The total number of subscription commands sent from clients to the server.
Total Devices	The total number of devices involved in synchronization. This total includes the same user multiple times if he or she has multiple devices. The value comprises individual devices, and does not count multiple synchronizations requested by the same device.

KPI	Description
Average/Minimum/Maximum Message Processing Time	The average, minimum, or maximum amount of time Unwired Server took to respond to a synchronization request message.
Time at Minimum/Maximum Processing Time	The time at which the shortest or longest response time completed.
Total Inbound Messages	The total number of messages sent from clients to Unwired Server.
Total Outbound Messages	The total number of messages sent from Unwired Server to clients.
Total Operation Replays	The total number of operation replays performed by clients on mobile business objects (MBOs).
Total Errors	The total number of errors that took place across all synchronizations.
Total Data Push	The total amount of data transmitted from the server to clients.

User Statistics

User statistics consist of key performance indicators that reflect the overall activity of application users.

User statistics can be filtered to include users who belong to a particular security configuration. This type of monitoring highlights key totals and identifies average, minimum, and maximum values for primary user activities. These calculations are dynamic, and are based on the data currently available in monitoring database for the specified time period.

Note: These statistics are not supported for Sybase Mobile CRM and Sybase Mobile Workflow for SAP application users.

Replication User Statistics

Replication user statistics reflect the synchronization activity of a group of replication-based synchronization users belonging to a specified security configuration. These statistics include general activity-related information on a per-user basis.

These key indicators monitor replication users:

KPI	Description
Total Synchronization Requests	The total number of sync requests initiated by a client.
Total Rows Received	The total number of rows received during package synchronization.
Total Rows Sent	The total number of rows sent during package synchronization.
Total Bytes Received	The total number of bytes uploaded from clients to Unwired Server.

KPI	Description
Total Bytes Sent	The total number of bytes downloaded by clients from Unwired Server.
Average/Minimum/Maximum Synchronization Time	The average, minimum, or maximum amount of time Unwired Server took to complete a synchronization request.
Time at Maximum/Minimum Synchronization Time	The time at which the fastest or slowest synchronization is completed.
Total Operation Replays	The total number of operation replays performed by user of mobile business objects (MBOs).
Total Errors	The total number of errors that took place across all synchronizations.
Total Devices	The total number of devices involved in synchronization. This total includes the same user multiple times if he or she has multiple devices. The value comprises individual devices, and does not count multiple synchronizations requested by the same device.

Messaging User Statistics

Messaging user statistics reflect the synchronization activity of a group of messaging-based synchronization users belonging to a specified security configuration. These statistics include general activity-related information on a per-user basis.

These key indicators monitor messaging users:

KPI	Description
Total Devices	The total number of devices involved in synchronization. This total includes the same user multiple times if he or she has multiple devices. The value comprises individual devices, and does not count multiple synchronizations requested by the same device.
Average/Minimum/Maximum Message Processing Time	The average, minimum, or maximum amount of time Unwired Server took to respond to a sync request message.
Time at Minimum/Maximum Message Processing Time	The time of day at which the shortest or longest message processing event completed.
Total Inbound Messages	The total number of messages sent from clients to Unwired Server.
Total Outbound Messages	The total number of messages sent from Unwired Server to clients.
Total Operation Replays	The total number of operation replays performed by clients on mobile business objects (MBOs).

KPI	Description
Total Errors	The total number of errors that took place across all synchronizations.
Total Subscription Commands	The total number of subscription commands sent from clients to the server.
Total Data Push	The total number of import data messages.

Security Log Statistics

The security log reflects the authentication history of users either across the cluster, or filtered according to domain, during a specified time period. These statistics allow you to diagnose and troubleshoot connection or authentication problems on a per-user basis.

User security data falls into these categories:

Category	Description
User	The user name
Security Configuration	The security configuration to which the device user belongs
Time	The time at which the authentication request took place
Result	The outcome of the authentication request: success or failure
Device ID	The device ID associated with the user
Package	The package the user was attempting to access
Domain	The domain the user was attempting to access

Cache Statistics

Cache statistics provide a granular view of cache activity eithr at the domain or package level, particularly in the areas of cache performance, mobile business object (MBO) status, and cache group status.

Cache statistics report on performance at the domain, package, MBO, and cache group levels to allow administrators to obtain different information according to the level of specificity required. These calculations are dynamic, and are based on the data currently available in monitoring database for the specified time period.

Note: These statistics are not supported for Sybase Mobile CRM and Sybase Mobile Workflow for SAP application users.

Viewing Package-Level Cache Statistics

Use the package tree to view cache statistics at the package, cache group, or mobile business object (MBO) level.

The package tree allows for a granular view of data in all cache statistic categories except for domain-level data. Domain-level data instead uses the Filter by Domain search functionality.

1. In the left navigation pane, select **Monitoring**.
2. In the right administration pane, select **Cache Statistics**.
3. From the cache feature drop-down list, select one of the following, depending on the type of data and level of granularity you require:
   - **Domain level**
   - **Package level**
   - **Package level cache group**
   - **Package level MBO**
4. Select **Show Package Tree**.
   The tree view appears on the left side of the right administration pane.
5. In the tree view, click the package, cache group, or MBO for which you want to view monitoring data.
   The monitoring data displays in the monitoring console. You can further filter data by specifying a time period in the search panel (for package-level cache performance and package-level MBO status only).

Cache Performance Statistics

Cache performance statistics report on key totals and identify average, minimum, and maximum values for primary cache activities. View cache performance data at the domain or package level.

Select either **Domain level** or **Package level** to view the following key performance indicators:

Key performance indicator	Description
Domain	The domain to which the package affected by the cache activity belongs.
Package	The name of the package associated with this cache activity.
Minimum/Maximum Cache Misses	The minimum or maximum number cache misses and the MBO name for which it was generated.
Minimum/Maximum Cache Hits	The minimum or maximum number of scheduled cache queries for all of the MBOs in the package in the specified date range.

Key performance indicator	Description
Minimum/Maximum/Average % Cache Hits	The minimum or maximum percentage of scheduled cache queries for the supplied date range and the MBO name for which it was generated.
Minimum/Maximum Average Wait Time	The minimum or maximum average wait time for a scheduled cache query and the MBO name for which it was generated.
Minumum/Maximum Average Refresh Time	(Package-level only)The minimum or maximum average refresh time for an on-demand or scheduled refresh.

MBO Statistics

Mobile business object (MBO) status monitoring reports on cache activity at the MBO level, and thus, reflects activity for single mobile business objects.

Select **Package level MBO** to view the following key performance indicators:

Key performance indicator	Description
Cache Group	The name of the group of MBOs associated with this cache activity.
MBO	The name of the single mobile business object associated with this cache activity.
Number of Rows	The number of rows affected by the cache refresh.
Cache Hits	The number of scheduled cache queries that occurred in the supplied date range.
Cache Misses	The number of on-demand cache or cache partition refreshes that occurred in the supplied date range.
Access Count	The number of cache queries that occurred in the supplied date range.
Minimum/Maximum/Average Wait Time	The minimum, maximum, or average duration of cache queries in the supplied date range. This time does not include the time required to refresh the cache in a cache “miss” scenario. Instead Minimum/Maximum/Average Full Refresh Time exposes this data.
Minimum/Maximum/Average Full Refresh Time	The minimum, maximum, or average duration of on-demand and scheduled full refresh activities in the supplied date range.

Cache Group Status Statistics

Cache group status statistics provide monitoring data about cache activity at the cache group level. The data reflects activity for all mobile business objects (MBOs) belonging to a cache group.

Select **Package level cache group** to view the following key performance indicators (KPIs):

KPI	Description
Package	The name of the package to which the associated cache group belongs
Cache Group	The name of the group of MBOs associated with the cache activity
Number of Rows	The number of rows in the cache table of the MBO
Last Full Refresh Time	The last time the cache or cache partition was fully refreshed
Last Update Time	The last time a row in the cache was updated for any reason (row-level refresh, full refresh, partitioned refresh, alternate read, or data change notification)
Last Invalidate Time	The last time the cache was invalidated
Cache Coherency Window	The data validity time period for the cache group, in seconds. Can span any value in this range: • 0 shows that data is always retrieved on-demand for each client. • 2049840000 shows that the cache never expires. This occurs when you set the on-demand cache group to NEVER expire or scheduled cache group to NEVER repeat.

Troubleshoot

Solve problems in Sybase Control Center.

Troubleshoot Sybase Control Center for Sybase Unwired Platform

Provides troubleshooting information for the Sybase Control Center (SCC) used to manage and monitor 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. Search *System Administration* for *System Diagnostics*.
4. Review device statuses – access the Device Users node in the left navigation tree of Sybase Control Center to view replication- and messaging-based synchronization device and user information.
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 at least 9.0.124 or higher. However, Sybase recommends that you use the most recent available version of Flash Player 10. For more information on software prerequisites, see the *Sybase Unwired Platform Installation Guide*.

Sybase Unified Agent Windows Service Fails to Start

Problem: When starting the Sybase Unified Agent 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_3-0\log\agent.log` shows the following message:

```
2009-11-10 13:35:50,752 [INFO ] [main]
com.sybase.ua.util.process.ProcessHandler.startProcess(395) -
Starting process SQLAnywhere-scc_repository: C:\Sybase
\SCC-3_0\services\SccSADataserver\sa\bin_windows32\dbsrv11.exe
-n scc_repository -o C:\Sybase\SCC-3_0\services\Repository
\scc_repository.slg -m -qi -qw -sb 0 -gn 100 -gm 500 -zl
-zp -x TCPIP{port=3638} C:\Sybase\SCC-3_0\services\Repository
\scc_repository.db
2009-11-10 13:36:57,037 [ERROR] [main]
com.sybase.ua.services.asa.ASAEngine.startEngine(135) - Failed to
connect to
dataserver engine. ASAEngine: Unable to open a new connection to the
engine.
```

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-3_0\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 Unified Agent service. Resolve this error by:

1. Creating a backup of `<UnwiredPlatform_InstallDir>\SCC-3_0\services\Repository\scc_repository.log`.
2. Deleting the `<UnwiredPlatform_InstallDir>\SCC-3_0\services\Repository\scc_repository.log` file and restarting the Unified Agent service.

Solution 2: Review `<UnwiredPlatform_InstallDir>\SCC-3_0\services\Repository\scc_repository.log` log for any failures in database recovery. Resolve this error by temporarily configuring the consolidated database to start without a transaction log:

1. Log out of Sybase Control Center.
2. Open `<UnwiredPlatform_InstallDir>\SCC-3_0\services\SccSADataserver\service-config.xml`.
3. Locate this line: `com.sybase.asa.server.options`.
4. Append "-f" to the value of this property.
5. Save the changes.

6. Restart the Unified Agent service.
7. Once the database server has processed the incomplete transaction, it will shutdown the database.
8. Confirm the successful transaction in the log file.
9. Revert the configuration of `<UnwiredPlatform_InstallDir>\SCC-3_0\services\SccSADataserver\service-config.xml`.
10. Restart the Unified Agent service.

Sybase Unified Agent Windows Service Deleted

Problem: the Sybase Unified Agent Windows service was inadvertently deleted, so Sybase Control Center is unavailable.

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

```
UnwiredPlatform_InstallDir\SCC-3_0\utility\ntautostart\release
\uaservice.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:
   a. Log out of Sybase Control Center.
   b. Stop the Sybase Unified Agent 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 Unified Agent service.
   g. 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*. 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 authentication configuration – ensure that the Sybase Control Center authentication provider configuration is correct, and points to the correct server. See *Security for Administration Users* in the *System Administration* guide for more information.
- Verify LDAP listener port consistency – if you are using LDAP security, ensure that the LDAP server port number is the same in all configuration files. The configuration files you may need to edit vary, depending on license type, or whether Unwired Platform shares the same repository as Sybase Control Center.
  - For development licenses: `<UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\UnwiredServer\OpenDS\config\config.ldif` and check the **ds-cfg-listen-port** property. This property value must match that specified in the Sybase Control Center authentication provider configuration file in addition to Unwired Platform if they share the same repository.
  - For Sybase Control Center authentication:`<UnwiredPlatform_InstallDir>\SCC-XX\conf\csi.properties`, and configuration should point to the correct LDAP host and port with appropriate properties.
- Ensure that Unwired Platform administration roles are correctly mapped to Sybase Control Center roles. Otherwise, you must first log in to Sybase Control Center as the sccAdmin (no password) and then, when you authenticate with Unwired Server, log in again with your Unwired Platform administrator role. For information about logging into Sybase Control Center and authenticating with Unwired Server, search for the Sybase Control Center online help for *Getting Started with Unwired Platform Administration*. For information about mapping roles, see *Security for Administration Users* in the *System Administration* guide.

- Check <*UnwiredPlatform_InstallDir*>\SCC_XX\log\agent.log for any issues with starting various services.

Cannot Access Unwired Platform Administration Nodes

Problem: Unwired Platform navigation nodes are documented but are either not visible or not functional in Sybase Control Center.

Solution: Ensure that you have the correct administration roles assigned, and that your product and version is Sybase Unwired Platform 2.0. Other products like Sybase Mobile Sales or Sybase Mobile Workflow may have visible Unwired Platform features, but may not have all of the same functionality as Sybase Unwired Platform 2.0.

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.

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.

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 *System Administration* 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.

Management Issues with Clustered Data Tiers

Problem: if you install Unwired Platform and the consolidated database on MS Cluster, you will receive errors when trying to manage the cluster in Sybase Control Center. This is because MS Cluster uses node switches.

Solution: Replace the current entry for the cluster with a new entry that uses the computer node's hostname or IP address, rather than Unwired Platform cluster's hostname (the default).

Platform Component Monitoring Issues

Review this list of documented issues for platform components monitored by Sybase Control Center.

Monitoring Data Does Not Appear in History Tab

Problem: Monitoring data does not appear immediately in the History tab.

Explanation: The monitoring data is stored in memory to optimize database access, and periodically flushed to the monitoring database.

Solution: Try either of these options:

- Wait for the data to be flushed. The default time period is five minutes.
- Change the flush interval to a smaller value in Sybase Control Center:
  1. In the left navigation pane, select **Monitoring**.
  2. In the right administration pane, select the **General** tab.

3. Click **Configuration**.
4. In the flush threshold section, ensure that **Enable flush threshold configuration** is selected.
5. Select one of:
   - **Number of rows** – monitoring data that surpasses the specified number of rows is flushed from the console display. Enter the desired number of rows adjacent to **Rows**. The default is 100.
   - **Time interval** – monitoring data older than the specified time interval is flushed from the console display. Enter the desired duration adjacent to **Minutes**. The default is 5.
   - **Either rows or time interval** – monitoring data is flushed from the console display according to whichever value is reached first: either the specified number of rows or the specified time interval. Enter the desired rows and duration adjacent to **Rows** and **Minutes**, respectively.
6. Retrieve the results list using the Sybase Control Center monitoring node.

Previously Existing Monitoring Data No Longer Appears

Problem: Monitoring data that displayed previously no longer appears.

Explanation: By default, monitoring data is preserved in the database for seven days. After that period, the data is removed.

Solution: Change the auto purge setting value in Sybase Control Center. Auto purge clears obsolete data from the monitoring database once it reaches the specified threshold.

1. In the left navigation pane, select **Monitoring**.
2. In the right administration pane, select the **General** tab.
3. Click **Configuration**.
4. In the auto purge section, ensure that **Enable auto purge configuration** is selected.
5. Enter the length of time (in days) to retain monitoring data before it is purged.
6. Restart the server.
7. Retrieve the results list using the Sybase Control Center monitoring node.

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-XX\services\EmbeddedWebContainer\log\http-service.log` (the HTTP access log).

  Solution:

  1. Close Sybase Control Center.
  2. Stop Sybase Unified Agent Service.
  3. Delete the contents of: `<UnwiredPlatform_InstallDir>\SCC-XX\services\EmbeddedWebContainer\container\Jetty-6.1.22\work`.
  4. Restart Sybase Unified Agent 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:

- The security repository (LDAP server) is down or not reachable. In the default configuration (excluding the Deployment Edition), the security repository is the OpenDS LDAP server.

  Solution: Check the status of the security repository to ensure that the server is reachable. For the OpenDS LDAP server, check the Services in the Windows Control Panel to ensure that the service is running and that the server is reachable from Sybase Control Center and the Unwired Server host.
- 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**.
- You do not have the required administration privileges.

  Solution: Ensure that:

  1. The administrator login is assigned one of the physical roles attached to the "SUP Administrator" or "SUP Domain Administrator" logical roles in the "admin" security configuration. To change the "admin" role mapping outside of Sybase Control Center, manually edit `<UnwiredPlatform_InstallDir>\UnwiredPlatform`

`\Servers\UnwiredServer\Repository\CSI\conf\role-mapping.xml` on all nodes in the cluster.

2. The `<UnwiredPlatform_InstallDir>\SCC-XX\conf\roles-map.xml` configuration for the "SUP LDAP Login Module" and your security repository are being used to authenticate Sybase Control Center users. Ensure that the administration roles configured for the "SUP LDAP Login Module" exist in the security repository.

- 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.

Section 4: 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 UAF agent windows service. After stopping the UAF 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.

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 Devices and ADMIN_WEBSERVICE_INVOCATION_ERROR in Agent.log

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

If users report problems synchronizing from the client, check for this error message in the Sybase Control Center agent.log file:

```
2009-11-10 17:19:51,718 [ERROR] [btpool1-4]
com.sybase.scc.jmx.ManagedObjectGateway.invoke(?) - Failed to invoke
managed object '82061EE7A2D141E391B46D245EAF0B1E' mbean operation
searchDeviceList2([java.lang.String]):
com.sybase.uep.sysadmin.management.mo.exception.SUPArtifactMOExcept
ion: 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: There are two possible reasons – the service could not be started due to a service account credential issue, or there is a conflict on the currently configured port for the administration web service.

If you install Sybase Unwired Platform with a domain account with administrator rights, and subsequently change the password, or use a domain account that does not have administrator rights, the Sybase Messaging Service will not start. One way to verify this is by accessing the following URL from the host where Sybase Unwired Platform is installed: http://localhost:5100/MobileOffice/Admin.asmx. The default Messaging Service port is 5100, but this may vary depending on your configuration.

Solution 1: Update or provide the username and password:

1. Navigate to **Control Panel > Administrative Tools > Services.**
2. Right-click Sybase Messaging Service, and select Properties.
3. Select the Log On tab, then update or provide the username (with administrative rights), and correct the password.

Solution 2: Make sure the administration Web service is up and running, and correctly configured. Review *Troubleshooting > Troubleshoot the System > Troubleshoot the Installation > Cannot Access Devices Tab and Web Service Error* topic on how to configure the port in case of conflict on 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 Unified Agent 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 Unified Agent 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.

Port Conflict Issues

Problem: You have identified a Sybase Unified Agent port conflict.

Solution:

1. Identify the service with the port conflict in `<UnwiredPlatform_InstallDir>\SCC_3-0\log\agent.log`.
2. Use a text editor to open `<UnwiredPlatform_InstallDir>\SCC_3-0\Services\<Servicename>\service-config.xml`.
3. Change the port to an available port number.
4. Save and close the file.

See *Sybase Unwired Platform System Administration Guide > System Reference > Port Number Reference* for more information.

Unexpected Listener Startup or Connection Errors

Problem: You encounter unexpected listener startup or connection errors for Unwired Platform components.

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 prevent caching of login credentials or reduce the cache period, decrease the default login cache timeout by changing the loginCacheTimeout=60 property in `<SUP_HOMW>/Servers/UnwiredServer/Repository/Instance/com/sybase/djc/security/SecurityDomain/default.properties`. Repeat this change on all server nodes in the cluster.

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.

Increasing Messaging Queue Counts Degrades Performance

Problem: Both inbound and outbound messaging queue counts were increased, however, performance degraded as a result.

Description: After increasing inbound and outbound message queue count, the default maxThreads of IIOP socket listener is insufficient.

Solution: Increase the maxThreads of IIOP socket listener by editing the `<hostname>_iiop1.properties` file (located in `<UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\UnwiredServer\Repository\Instance\com\sybase\djc\server\SocketListener\`), and restart Unwired Server. The maxThread of IIOP socket listener must be larger than the sum of all nodes needed IIOP thread counts.

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.

Package Deployment and Management Issues

Review this list of documented issues for packages deployed or managed from Sybase Control Center.

Exporting or Deploying Large Packages Fails

Problem: You used Sybase Control Center to export or deploy a large package, and it fails.

You can troubleshoot this error by opening the Sybase Control Center `<UnwiredPlatform_InstallDir>\SCCXX\log\agent.log` file and checking for a message that is similar to this one:

```
exception:java.lang.IllegalStateException: Form too large
```

Explanation: This message means that the package, and not the form, is too large. The Web server that hosts Sybase Control Center cannot manage the data. A number like 273310 indicates the size of the package in kilobytes (that is, 273,310).

Solution 1: Use this solution if you run the Sybase Unified Agent as a service (default).

1. Sybase recommends you create a backup copy of the Windows registry before proceeding.
2. Open the registry editor by running **regedit** at the command line.
3. Locate the Sybase Unified Agent 3.0 registry key: HKEY_LOCAL_MACHINE \SOFTWARE\SYBASE\Unified Agent 3.0.
4. Set the `jvmopt1` property to a value larger than the default, and save the file. The default is 2000000. For example:
   ```
   -Dorg.mortbay.jetty.Request.maxFormContentSize=2048000
   ```

Solution 2: Use this solution if you do not run the Sybase Unified Agent as a service.

1. Close Sybase Control Center, and stop Sybase Unified Agent using the Windows Services dialog.
2. Open the `uafstartup.bat` file, located at `<UnwiredPlatform_InstallDir>\SCC_3-0\bin\uafstartup.bat`, in a text editor.
3. Set the **maxFormContentSize** property to a value larger than the default, and save the file. The default is 2000000. For example:
   `-Dorg.mortbay.jetty.Request.maxFormContentSize=2048000`

4. Restart the Sybase Unified Agent (using the updated `uafstartup.bat` file), and reopen Sybase Control Center.

Invalid DOE-C User Error for an SAP Server Connection

Problem: The General tab of a DOE-C package displays an invalid user account error for the Error State property.

Explanation: SAP servers could not authenticate this user with the Username and Password configured for this package.

User names and passwords configured for the connection pool cannot be tested before they are used. Errors are only reported after the connection fails. Errors typically occur during an administrative operation (such as unsubscribing a subscription), or in response to an asynchronous message for a subscription from DOE. On a system with existing DOE-C subscriptions, the initial resynchronization at startup would implicitly test the technical user.

Solution: Check the username and password configured for this user in the Connection Pool configured for the package. If it is incorrect, edit the properties used.

Note: If you change the username or password property of a DOE-C connection, you must reopen the same dialog and click `Test Connection` after saving. Otherwise the error state of this DOE-C package cannot be cleaned up. If you do not click `Test Connection`, the username or password is correct, but the error state of the DOE-C package cannot be cleaned up.

Device and Device User Management Issues

Review this list of documented issues for devices or device users managed by Sybase Control Center.

Wrong Device for Code Error

Problem: Device registration using a Windows Mobile emulator appears successful in Sybase Control Center, but the device log shows a `Wrong Device for Code` error when the device 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 device IDs, the Unwired Server messaging software on the device creates a device ID during installation and stores it in the emulator device registry. After registration, this permanent link between the emulator and the device 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 device ID for the device. Without the original device ID, the server cannot identify the device emulator, and therefore, cannot establish a relationship between the device 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 device from Unwired Server, then reregister the device, or
2. Reregistering the device.

User Name of Registered Device Not Displayed

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

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

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

1. In the left navigation pane, click the **Device Users** node.
2. In the right administration pane, click the **Devices** tab.
3. In the table of registered devices, select MBS to display the **Activation User Name** column and look for the device ID.

Internal Server Error When Clicking Device User

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

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

Solution:

1. Validate the error:
   a. Open <*UnwiredPlatform_InstallDir*>\SCC-3_0\log\agent.log.
   b. 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 Messaging Server service (obmo.exe) is running.
3. Validate that the Messaging Server Administration Web Service is running:
   a. Open a Web browser.
   b. Open http://localhost:5100/MobileOffice/admin.asmx.
   c. Select the **GetDeviceList2** method, then click **Invoke**.
   d. 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:
   a. Restarting the Messaging Server service.
   b. Once available, repeat steps 2-3.
      - If you still do not yield an expected result, then reinstall the Unwired Server.
      - Otherwise, open Sybase Control Center, and click Device User to try registering a user again.
5. If you still get the same error and same behavior, contact Sybase Support. Ensure you:
   - Have a copy of the Messaging Server deviceManagement trace file.
   - Including the information on which specific Admin WS method is called that causes the error, and what parameter value is used.

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 seperately 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.

APNS – Apple Push Notification Service.

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

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 consolidated database that store synchronization data. *See* CDB.

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

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 parameter used by the operation to load data into the cache from the enterprise information system (EIS).

CDB – Consolidated database. The CDB 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 for BlackBerry*, *Developer Guide for iOS*, and *Developer Guide for Windows Mobile*.

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 consolidated, 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.

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 consolidated database through a push event.

data refresh – A data refresh synchronizes data between the consolidated 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 consolidated database (CDB), cluster, monitoring, and messaging databases.

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-based synchronization (RBS) 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 RBS 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 requriement 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 (MBS) – A synchronization method where data is delivered asynchronously using a secure, reliable messaging protocol. MBS 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* replication based 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 consolidated, 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 message-based 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-based synchronization, messaging-based 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 consolidated 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.

openDS – The default LDAP server that is installed in Developer Edition and is suitable for authentication and authorization in a development environment.

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) 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 parameters – perform a data refresh based on the value of the parameter.
- 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: Session, 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.

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 CDB. On Windows Mobile, pull synchronization is supported only in RBS 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.

replication based synchronization (RBS) – A synchronization method where data is delivered synchronously using an upload/download pattern. For push-enabled clients, RBS uses a "poke-pull" synchronization 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.

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 Serverand 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.

scheduled data refresh – Data is updated in the consolidated 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 and monitor the environment. You can also deploy MBO packages and manage deployed MBO packages in order to design the synchronization behavior for those packages. 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 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 Unified Agent – Provides runtime services to manage, monitor, and control distributed Sybase resources. The agent must be running for Sybase Control Center to run.

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 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 consolidated database) or download event (from the consolidated 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.

Visual SQL – A graphical user interface tool that you can use to build SQL queries.

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.

Glossary: Sybase Control Center

Defines general Sybase Control Center terms.

alert – A mechanism for notifying administrators when a managed resource experiences a status change or when a performance metric passes a user-specified threshold.

alert instance – A copy of an alert template that, when enabled, generates a particular kind of alert.

alert notification – A message sent to a target (such as an email address) as the result of an alert.

alert target – The destination for an alert notification and source of an alert response.

alert template – Generic definition of a specific event that can generate notifications. See also alert instance.

alert type – A template used to define special processing for alerts.

alert storm – The result of issuing many redundant alerts associated with a common or root occurrence.

alert suppression – The suppression of redundant alerts resulting from an alert storm.

availability – Indicates that a resource is accessible and responsive.

collection – A named, predefined set of key performance indicators whose values are collected from monitored servers at the same time. Collections supply the performance and availability data shown on Sybase Control Center screens and charts. Use the scheduler to view a list of collections and to control which collections run, how often they run, and for how long.

event – An activity in the system such as a user logging in, a service starting or stopping, or a condition changing. You can use the alerts feature to detect and notify you about system events.

heat chart – A graphical view of resource availability in the current perspective.

job – A task performed by the scheduler in Sybase Control Center.

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.

repository – A storage area that archives the performance statistics of your managed resources.

resource – A unique Sybase product component (such as a server) or a subcomponent.

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.

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.

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

T

U

V

W