SYBASE®
An SAP® Company

Administration Workbook

Sybase Unwired Platform 2.2

DOCUMENT ID: DC01625-01-0220-01
LAST REVISED: October 2012

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

Contents

How to Use this Workbook

This workbook assists Unwired Platform administrators understand what steps are required in implementing different production deployments. Assume the personas and understand the scenarios presented in this document, so that you can follow the workflows as required.

This workbook contains:

- Business Requirements – the relevant Unwired Platform business requirements that apply to the activity.
- Activity – a workflow you can follow for training or testing purposes.

Learning Activities, Goals, and Co-Requisites

This workbook is intended to be used in tandem with existing Sybase® Unwired Platform documentation.

Workbook activities are modular, thereby allowing you to mix-and-match various activities according to your individual requirements. As you perform a workbook activity, you may want to have read or use the documents identified in the co-requisites column:

Workbook activity	Goal	Co-requisites
Activity 1: Setting Up a Highly Available and Secure Development Environment	Create a single-node Relay Server-enabled cluster for a development/test environment.	Read: • (Recommended) *> Adding Relay Servers or Reverse Proxies* in *Landscape Design and Integration* • (Recommended) *Security > DMZ Security*

Workbook activity	Goal	Co-requisites
Activity 2: Testing Package Deployment on a Test Domain	Set up multiple domains and use them to support different tenants.	Complete: • (Required) *Activity 1: Setting Up a Highly Available and Secure Development Environment* Read: • (Recommended) *System Administration > Domain Management > Enabling a Multitenancy Environment with Domains* • (Recommended) *System Administration > Domain Management*
Activity 3: Automate Application Connection Registration	Use the Sybase Administration API to automate the registration of application connections.	Complete: **1.** (Required) *Activity 1: Setting Up a Highly Available and Secure Environment* **2.** (Optional) *Activity 2: Testing Application Deployment on a Test Domain* Read: • (Recommended) *System Administration > System Reference > Command Line Utilities > Unwired Server Runtime Utilities > Package Administration (supadmin.bat) Utility* Refer: • *Developer Guide: Unwired Server Runtime > Management API*

Usage Scenario

This scenario describes the post-installation development and test environment setup process. Assume the role of actors as they perform key activities for this environment type.

John just installed Sybase Unwired Platform for a development and test environment for research and development teams for Acme Corporation. Acme Corporation is a mobility enablement company, specializing in mobile solutions for mid-size clients. One of its largest

clients, ABC corporation, requires both applications and solutions hosting. This workbook describes the relationship between these organization as well as describes the actors that perform administration tasks for each company.

Actors in this Workbook

An actor is a combination of a user and a role. This workbook requires that you perform an activity from the perspective of a specific actor. This perspective allows you to understand how tasks interrelate in a given deployment of Unwired Platform.

The environment:	Acme Corporation purchased Unwired Platform. The organization builds mobile applications and offers mobile hosting services to its customers. Initially, Acme will set up a development and test environment, then scale the deployment of Unwired Platform to support domains.
Acme users and roles:	• John is the platform administrator. • Jane is the security administrator. • Tom develops mobile applications.
The client:	ABC Corporation is a customer of Acme Corporation.
ABC users and roles:	Ram is the domain administrator.
End user:	Mary is the application tester for the device applications that Tom builds.

Activity 1: Setting Up a Highly Available and Secure Development Environment

Goal: Install and configure Unwired Server as a relay server-enabled development cluster, and configure the platform components to communicate with it.

In a development environment, you can deploy a relay server to:

- Make the Unwired Server runtime highly available to devices.
- Help balance load among multiple Unwired Server nodes, thereby making the runtime more fault tolerant.

See also

- *Technical Prerequisites* on page 16

Business Requirements for Setup

Business requirements for setup span two organizations, multiple policies, several roles.

Requirement	Actions required
Security policy dictates that all users authenticate with Unwired Platform using existing corporate login credentials.	John must use Acme's enterprise security repository for internal users. Acme employees have a valid account and are assigned group memberships as required. Need to coordinate roles for: • SUP Administrator – John self-delegates this role to his existing user profile. • SUP Developer – John sets up Tom for this role, so that Tom can deploy applications and use the system for testing. • SUP User – John sets up Mary for with this role, so she can test Tom's application.
Security policy requires all default passwords be changed.	John needs to change the cache database (CDB) password used by Unwired Platform by default to one that is unique for Acme, if John did not change it during installation.
ABC Corporation is a client that requires hosted mobility services. Domain-level security must be configured accordingly.	John must set up Ram as domain administrator of ABC's domain and artifacts, and create a security configuration to authenticate domain users requesting access to these domain artifacts.
Acme's quality assurance policy dictates that the testing environment closely replicate the production environment, but still be cost-effective (that is, not require excessive amounts of hardware purchases).	John must: • Configure the Unwired Servers and create a two-node server cluster to replicate performance requirements of the production environment, but keep the environment cost-effective. • Enable testing application from outside of the firewall, by using Relay Server and configuring Relay Server Outbound Enablers (RSOEs) to use this service during access requests.
Cache updates must be through data change notifications (DCNs) from in-house systems, because of a requirement to support delivery over a secure interface.	John needs to configure Unwired Server to use DCN to update the cache according to production system requirements.

Technical Prerequisites

Understand the prerequisites for this activity.

This activity requires that John:

- Install Unwired Platform with the correct development license. For details, see *Landscape Design and Integration > Stage 2: Design > Choosing Licenses > Assessing License Needs > License Types*.
- Deploy Relay Server on an IIS 7.x host. For details, see *Stage 3: Implement* in *Landscape Design and Integration*.
  - *Manually Installing Relay Server on IIS*
- Has access to installation of an LDAP server and browser.

Development Environment Setup Task Flow

Setting up and configuring a highly available development environment so it communicates securely over wired and wireless networks requires a conjoint effort between John and Jane.

Securing the Administration Infrastructure

Jane updates the LDAP directory, and John secures the administration components. Together they help to secure the administration infrastructure.

1. *Preparing the LDAP Directory*

   As security administrator, Jane modifies the OpenDS LDAP repository to set up required user and group accounts.
2. *Configuring Unwired Server Security*

   As Platform administrator, John now secures the Unwired Server components.

Preparing the LDAP Directory

As security administrator, Jane modifies the OpenDS LDAP repository to set up required user and group accounts.

Jane creates user accounts for Tom, Mary, and John, and grants membership to appropriate Unwired Platform groups:

- Tom becomes a member of Acme's "SUP Developer" group (which allows him to deploy application packages to Unwired Server as required).
- John becomes a member of the Acme "SUP Administrator" group.
- Mary becomes a member of Acme's “SUP User” group.

1. *Use an LDAP Browser to Import Encoded Passwords From LDIF*

   Because Acme has an LDIF file that includes encoded passwords, Jane must configure an LDAP browser to handle this correctly on import.
2. *Connecting Apache DS to OpenDS*

   Jane uses Apache DS as the LDAP browser.
3. *Importing LDIF Contents*

   Jane uses an LDAP Browser view to import LDIF file contents.

See also

- *Configuring Unwired Server Security* on page 8

Use an LDAP Browser to Import Encoded Passwords From LDIF

Because Acme has an LDIF file that includes encoded passwords, Jane must configure an LDAP browser to handle this correctly on import.

1. From the Windows Control Panel **Services** window, stop the LDAP server.
2. In a text editor, open <OpenDS_InstallDir>\config\config.ldif.
3. Change the value of the `ds-cfg-allow-pre-encoded-passwords` property to `true`.
4. Save the changes, then restart the LDAP server.

Connecting Apache DS to OpenDS

Jane uses Apache DS as the LDAP browser.

Before she can use Apache DS, Jane connects Apache DS Studio to OpenDS Server.

1. Launch Apache DS Studio: `<ApacheDS_InstallDir>\studio\Apache Directory Studio.exe`.
2. Add a connection to the OpenDS LDAP Server:
   a) Right-click the Connections view, then click **New Connection**.
   b) Configure these values:

Page	Property	Value
Network Parameters		
	Host	localhost
	Port	10389
	Encryption method	No encryption
Authentication		

Page	Property	Value
	Authentication method	Simple Authentication
	Bind DN	cn=Directory Manager
	Bind password	secret
	Save password	checked

c) Click **Check Authentication** to validate these properties.

3. If the operation is successful, click **Next** , then click **Fetch Base DNs** and click **OK**.
4. To connect, click **Finish**.

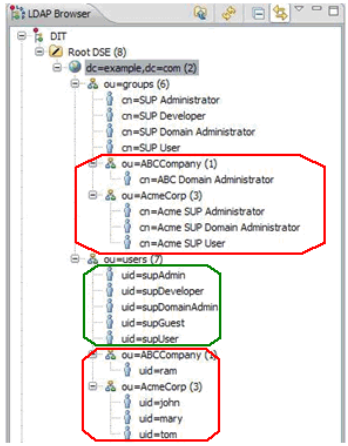
Importing LDIF Contents

Jane uses an LDAP Browser view to import LDIF file contents.

The LDIF file Jane imports has user accounts and encrypted passwords for John, Tom, Mary, and Ram. For the sake of this exercise, the password for each user is same as their login name.

1. In Apache DS Studio, switch to the LDAP Browser view.
2. Import the LDIF file by right-clicking the root of the Unwired Platform LDAP tree, then clicking **LDIF Import**.
3. Use the wizard to browse and open the LDIF file.
4. Verify that user accounts and groups are imported and added to the appropriate organization unit entry for both ABC Corporation and Acme Corporation.

   The green circle indicates existing platform users in the LDAP directory. The red circle indicates import changes to the directory data:



Configuring Unwired Server Security

As Platform administrator, John now secures the Unwired Server components.

John launches Sybase Control Center from the desktop shortcut. John uses Sybase Control Center to configure Unwired Server to use the LDAP directory updated by Jane. Because the LDAP directory is not yet set up to authenticate platform components, John logs in with the administration login credentials indicated during installation.

1. *Logging in as Platform Administrator*

   John logs in to Sybase Control Center to access Unwired Server.
2. *Configuring the Admin Security Configuration*

   John modifies the "admin" security configuration to use an LDAP directory.
3. *Mapping Default Platform Roles*

   John must map the Acme user roles defined by Jane in the LDAP directory (physical roles) to the Unwired Platform administration roles (logical roles). These mappings become default mappings for domain-level logical roles.

See also

- *Preparing the LDAP Directory* on page 5

Logging in as Platform Administrator

John logs in to Sybase Control Center to access Unwired Server.

1. In Sybase Control Center, enter the credentials indicated during installation.
2. Click **Login**.

John sees the full view of Unwired Platform.

Configuring the Admin Security Configuration

John modifies the "admin" security configuration to use an LDAP directory.

1. In the navigation pane of Sybase Control Center, expand the **Security** folder, then click the security configuration named **admin**.
2. In the administration pane, click the **Authentication** tab.
3. Click **New** to add an LDAP provider.
4. Select **com.sybase.security.ldap.LDAPLoginModule** as the login module.
5. Set following properties (assuming you are using LDAP running on localhost at port 10389.

Property	Value
controlFlag	optional
BindDN	cn=Directory Manager
BindPassword	secret
AuthenticationSearchBase	ou=users,dc=example,dc=com
DefaultSearchBase	dc=example,dc=com
ProviderURL	ldap://localhost:10389
RoleMemberAttributes	uniquemember
RoleSearchBase	ou=groups,dc=example,dc=com
ServerType	openldap
RoleScope	subtree
AuthenticationScope	subtree

6. Click **OK**.
7. In the **General** tab, click **Validate**.
   If the validation returns no errors, click **Apply** to save the configuration.

Mapping Default Platform Roles

John must map the Acme user roles defined by Jane in the LDAP directory (physical roles) to the Unwired Platform administration roles (logical roles). These mappings become default mappings for domain-level logical roles.

1. Expand the default domain node under the **Domains** folder, then expand the **Security** folder, and click the security configuration named **admin**.
2. In the **Role Mappings** tab, select **Map Roles** from the drop-down lists under **Physical Roles**.

Map this logical role	To this physical role
SUP Administrator	Acme SUP Administrator
SUP Domain Administrator	Acme SUP Domain Administrator

3. Click **OK**.
4. Close the Sybase Control Center broswer session.

Tuning Unwired Server Performance

John tunes Unwired Server performance properties to test the configuration and ensure it meets the performance requirements needed for a development environment. John uses Sybase Control Center to perform all these tasks. As done before, John launches Sybase Control Center URL from the desktop shortcut

Knowing that his user account has been granted platform administration privileges via the group he is a member of, John administers Unwired Platform with correct Platform administrator permissions.

1. *Configuring Server Performance Properties*

   Once John logs in as Platform administrator, he configures the JVM heap size to improve Unwired Server performance in a development cluster.
2. *Configuring RBS Performance Properties*

   John enhances replication-based synchronization (RBS) by tuning synchronization properties.
3. *Configuring CDB Connection Properties*

   John changes the maximum pool size for default connections.
4. *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.

Configuring Server Performance Properties

Once John logs in as Platform administrator, he configures the JVM heap size to improve Unwired Server performance in a development cluster.

1. Expand the **Servers** folder, then expand the primary Unwired Server name in your development cluster.
2. Click **Server Configuration**.
3. In the **General** tab, change the **Minimum Heap Size** and the **Maximum Heap Size** properties to `512M`.
4. Click **Save**. If there are no errors, changes are written to the configuration file.
   A confirmation message is displayed on top of the administration pane. A red error box indicates that the value is invalid.

Configuring RBS Performance Properties

John enhances replication-based synchronization (RBS) by tuning synchronization properties.

1. In the navigation pane of Sybase Control Center, click the **Configuration**.
2. In the **General** tab, click **Performance**.
3. Change the **Synchronization cache size** property to `80M` and **Thread Count property** to `50`.
4. Click **Save**. If there are no errors, changes are written to the configuration file.

Configuring CDB Connection Properties

John changes the maximum pool size for default connections.

1. Expand the **default** domain folder, then click **Connections**.
2. In the **Connections** tab, check the **default** connection in the Connection Pool Name column, then click **Properties**.
3. For the Max Pool Size property, change the value to 200.
4. Click **OK**.
5. When prompted for confirmation to change system database, click **Yes**.
   A confirmation message is displayed on top of the administration pane. If there is a red error box that indicates that the value is invalid.

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.

Preparing for Connections Outside the Firewall

Relay Server allows Mary to test RBS device clients in a development test environment by using wireless connections to Unwired Server from outside of the firewall. Relay Server enables these types of secure inbound server connections.

Jane has already installed Relay Server on a host that runs IIS and situated in the DMZ. John registers and configures Relay Server and RSOEs in Sybase Control Center, and exports the configuration files so this configuration can eventually be used in a production environment.

1. *Enabling Secure Device Connections by Deploying Relay Server*

   John configures the Relay Server in Sybase Control Center to create two farms for a single development and test cluster: one for RBS client connections and one for MBS client connections. John sets up both farms now, even though RBS connections are initially tested.
2. *Setting Up the RSOE for Unwired Server RBS Services*

   John uses Sybase Control Center to configure and start the RBS RSOEs for the development version of Unwired Server.
3. *Development Impact*

   Upon completion of the setup and deployment of both the Relay Server and RSOE, Tom can develop RBS device applications that Mary uses to determine whether the connection is viable for a production environment.

Enabling Secure Device Connections by Deploying Relay Server

John configures the Relay Server in Sybase Control Center to create two farms for a single development and test cluster: one for RBS client connections and one for MBS client connections. John sets up both farms now, even though RBS connections are initially tested.

1. In the navigation pane of Sybase Control Center, click the development cluster name.
2. Click the **Relay Servers** tab.
3. Click **New**.

4. Configure these general Relay Server properties:

Property	Value
Host	relay.acme.com
HTTP port	80
HTTPS port	443
(Optional) URL suffix	/ias_relay_server/server/rs_server.dll

5. Click **Next**.
6. Define the RBS farm:

Property	Value
Farm ID	SUPRBSFarm
Type	Replication

7. Click the + button.
8. In the bottom of the Relay Server Configuration window, assign the RBS farm to a node so that devices with RBS requests can connect to the Relay Server.
   a) Configure these properties:
      - **Node ID –** PrimarySUPDev
      - **Token –** abcd
   b) Click the + button.
9. Click **Finish**.
10. Generate the configuration to transfer the properties from the cluster database to the Relay Server host machine.
    a) Select the created Relay Server configuration and click **Generate**.
    b) Choose **Relay server configuration file** and output the whole file by selecting **Whole relay server**.
    c) Click **Next**.
    d) Click **Finish** and transfer the generated file to the corresponding Relay Server host computer. Update the existing `rs.config` file, and run `rshost.exe`to update the Relay Server configuration.

       For more information, search for these topics:
       - *Relay Server Configuration (rs.config)* in *System Administration*
       - *Generating and Modifying the Relay Server Configuration File* in *Landscape Design and Integration*.

Setting Up the RSOE for Unwired Server RBS Services

John uses Sybase Control Center to configure and start the RBS RSOEs for the development version of Unwired Server.

John must also load the correct certificate file because the connections will be secured using HTTPS.

Tip: If the certificate used by the IIS/Apache server (hosting the Relay Server plug-in) is issued by a well-known Certificate Authority, then this certificate configuration step is not required. Alternatively, you can choose to have RSOE connect to the HTTP port of IIS/Apache (device application clients will still connect to the HTTPS port of IIS/Apache).

1. In the navigation pane, click **Servers** > **<ServerNode>** > **Server Configuration**.
2. Select the certificate:
   a) In the administration pane, select the **Outbound Enabler** tab, then click **Certificate Files**.
   b) Click the **+** button to browse and select the `.CRT` file to upload, then click **Open**.

      Select the server public certificate used for securing the HTTPS port.
   c) Click **OK**.
3. Configure the RSOEs.

   While Sybase recommends three RSOEs per synchronization type, one should be sufficient for a development environment.
   a) In the administration pane, expand the **Servers** folder, expand the server name node, click the Server Configuration node, click the **Outbound Enabler** tab, then click **New**.
   b) Configure general properties:

Property	Value
Farm type	Replication
Unwired sever port	2480
Relay server host	relay.acme.com
Relay server port	443
Unwired server farm	SUPRBSFarm
Server node ID	PrimarySUPDev
Certificate file	The file just uploaded
Trusted certificate	Not applicable. The certificate file contains only a single certificate.

   c) Configure start-up options by accepting all defaults, and exit the wizard.
4. In the **Outbound Enabler** tab, select the newly configured RSOE and click **Start**.

Development Impact

Upon completion of the setup and deployment of both the Relay Server and RSOE, Tom can develop RBS device applications that Mary uses to determine whether the connection is viable for a production environment.

In the device application Tom is responsible for configuring the connections values so they exactly match those that John configured for the Relay Server-enabled environment. See the *Developer Guide* that corresponds to the device platform.

Activity 2: Testing Package Deployment on a Test Domain

Goal: Set up a domain and deploy a test package to it.

This scenario takes you through the steps of creating a domain. Initially Acme implements domains as a logical container to isolate and separate client packages, backend server connections, security configurations, and role mappings used by ABC (or potentially, other future clients). In the future domains Acme may use domains internally as well:

- Support Acme's IT environment by separating internal organization applications for different business units.
- Support Acme's development environment by separating development by teams that share the same Unwired Server.

Business Requirements for Domain Package Deployment

Business requirements are primarily driven by ABC via the service level agreement (SLA) they share with Acme.

Requirement	Action required
ABC Corporation is a client and needs hosted mobility services.	Because Ram is domain administrator for ABC, Ram must deploy the package to the correct domain. Furthermore, ABC expects authentication providers to be defined exclusively for its Unwired Platform domain.
Quality assurance policy dictates that new or modified applications be tested on the development environment first before deploying it to the production network on ABC's domain.	Packages that pass testing must be deployed to the production system easily, using the configuration defined with in the test cycle. Both Tom and John conjointly decide on those values. The test cycle can and should be used to enhance the mobility experience of its users if an update is required.

Requirement	Action required
ABC events must be logged to a domain-specific log and each event cannot be deleted until 10 days have passed. Furthermore, application usage must be monitored and performance indicators gathered to assist with ongoing environment tuning.	John must enable logging and monitoring at the domain level, and configure data retention according to the service level agreement defined.

Technical Prerequisites

Understand the prerequisites for this activity.

This activity requires that administrators:

- Set up a highly available and secure development environment.
- Ensure Unwired Server and Sybase Control Center services are started.
- Ensure the deployment package ready for testing.
- Ensure the deployment archive has these settings:
  - A name of `SUP101.jar`.
  - A cache group (with a cache interval set to 5 minutes) and a synchronization group (using the default — a change detection interval of 10 minutes).

Multitenancy Setup and Domain Deployment Task Flow

Creating a new domain and deploying a test application is an activity that is coordinated among John, Tom, and Ram.

In the immediate term, Acme's domain implementation must exclusively support development testing. Testing gives John and Tom the adjustments required to tune the production environment and package version.

1. *Setting up a Tenant Domain*

   John sets up a tenant domain so that packages can be hosted on Acme's Unwired Platform system.
2. *Deploying the Application Package for Testing*

   John deploys an application package for testing. During testing, John ascertains which development-cycle properties need to be updated with production-ready values.
3. *Importing the Package Archive on a Production Server*

   John exports the package from the default domain, then imports the package on a production Unwired Server to deploy it without needing to reconfigure the package.

Setting up a Tenant Domain

John sets up a tenant domain so that packages can be hosted on Acme's Unwired Platform system.

1. *Creating a Tenant-Specific Security Configuration*

   John creates a new security configuration that defines production-ready security providers for ABC to prepare the domain for eventual package deployment.
2. *Creating a Tenant Domain and Assigning a Default Provider*

   John creates a domain named ABC for and assigns the security configuration created exclusively for this tenant's domain.
3. *Defining and Assigning a Domain Administrator*

   John registers the domain administrator login and assigns two users (Ram and Tom) to different domains.
4. *Enabling Domain Logging*

   John enables and configures domain logging, so data can be captured in the domain log database. Ram reviews captured events for his domain.
5. *Enabling Domain Monitoring*

   John enables monitoring on the ABC domain to monitor usage and share key performance indicator data as defined in the service agreement.

See also

- *Deploying the Application Package for Testing* on page 21

Creating a Tenant-Specific Security Configuration

John creates a new security configuration that defines production-ready security providers for ABC to prepare the domain for eventual package deployment.

The package uses a security provider created exclusively for ABC's domain. These security providers are configured against ABC's own repository so only ABC users are the ones authenticated in a production deployment of the package.

1. Log into Sybase Control Center with the user credentials of John/John.
2. Click **Security** in the navigation pane, select the **General** tab in the administration pane, and click **New**.
3. In **Create Security Configuration**, name the security configuration `ABCAppSecurity`, then click **OK**.
4. In the navigation pane, expand the **Security** folder and click the new security configuration node.
5. In the administration pane, click the **Authentication** tab then click **New** to add a new LDAP security provider.

6. Select `com.sybase.security.ldap.LDAPLoginModule` as the login module.
7. Configure the LDAP properties.

   The **AuthenticationSearchBase** and **RoleSearchBase** properties ensure that only users and groups in the ABCCompany organization unit are allowed access to ABC data.

Property	Value
BindDN	`cn=Directory Manager`
ControlFlag	`required`
BindPassword	`secret`
AuthenticationSearchBase	`ou=ABCCompa-ny,ou=users,dc=exam-ple,dc=com`
DefaultSearchBase	`dc=example,dc=com`
ProviderURL	`ldap://localhost:10389`
RoleMemberAttributes	`uniquemember`
RoleSearchBase	`ou=ABCCompa-ny,ou=groups,dc=exam-ple,dc=com`
AuthenticationScope	`onelevel`
ServerType	`openldap`
RoleScope	`onelevel`

8. Click **OK**.
9. In the **Authentication**, **Authorization**, and **Attribution** tabs, delete **NoSecLoginModule**, **NoSecAuthorizer**, and **NoSecAttributer**, respectively.
10. In the **General** tab, click **Validate**.

    A confirmation message is displayed upon success.
11. If the validation is successful, click **Apply**.

Creating a Tenant Domain and Assigning a Default Provider

John creates a domain named ABC for and assigns the security configuration created exclusively for this tenant's domain.

1. In the navigation pane, click **Domains**.
2. In the administration pane, click **New**.
3. Type `ABC` as the domain name.
4. Click **Next**.

5. Select **ABCAppSecurity** as the default security provider for the ABC domain.
6. Click **Next**.
7. Click **Finish**.

Defining and Assigning a Domain Administrator

John registers the domain administrator login and assigns two users (Ram and Tom) to different domains.

When these users log in to Sybase Control Center, they must be identified as domain administrator. Ram was already defined as a member of the "ABC Domain Administrator" role when the LDIF file was imported in to the security repository. In addition, John determines that Tom needs to access the default domain. Tom is also already a member of the "Acme SUP Domain Administrator" group.

Note: Mapping roles to allow administrator access in Sybase Control Center was configured in Activity 1 of this workbook.

---

1. Register the domain administrator with the cluster:
   a) In the navigation pane, click the cluster's **Security** folder, then in the administration pane click the **Domain Administrators** tab.
   b) Register Ram's credentials by clicking **New**, configuring Ram's administrator properties, then clicking **OK**. Repeat this action to add Tom's information.
2. Map domain administrator physical and logical roles in the **admin** security configuration for the **default** domain.
   a) In the navigation pane, expand the **Domains** folder, then click **default** > **Security** > **admin**.
   b) In the **Role Mappings** tab, click the adjacent cell in the Physical Roles column for the **SUP Domain Administrator** logical role, then select **Map Roles**.
   c) In the **Role Mappings** dialog, add **Acme SUP Domain Administrator** and **ABC Domain Administrator** roles to the Mapped Roles column.
   d) Click **OK**.
3. Assign Ram to the ABC tenant domain.
   a) In the navigation pane, expand the **Domains** folder and click the **ABC** > **Security** folder, then in the administration pane click the **Domain Administrators** tab.
   b) Click **Assign**, then choose Ram from the list and click **OK**.
4. Assign Tom to the default domain.
   a) In the navigation pane, expand the **Domain** folder, select **default**, select the **Security** folder, then in the administration pane click the **Domain Administrators** tab.
   b) Click **Assign**, then choose Tom from the list and click **OK**.

Enabling Domain Logging

John enables and configures domain logging, so data can be captured in the domain log database. Ram reviews captured events for his domain.

John configures domain log data retention to 10 days. This allows Ram to review application activities periodically.

1. Enable domain event logging:
   a) In the navigation pane, click **Domains > ABC > Log**.
   b) In the administration pane, click the **Settings** tab.
   c) Click **New**.
   d) In the **Profile Definition** dialog, complete the **Name** and **Description** fields with the proper value.
   e) Select the check boxes of **Package related**, **Security related**, **Application connections**, **Connections**. and **Payloads**.
   f) Select **Enable after creation**.
   g) Click **OK**.
2. Change the domain log data retention configuration:
   a) In the navigation pane, select **Domains**, **default**, and **Log**.
   b) In the administration pane, select the **Settings** tab.
   c) Click **Configurations**.
   d) Select **Enable auto-purge configuration**, and set the data retention to **10 days**.
   e) Click **OK**.
3. Review package-level logging events:
   a) In the navigation pane, click **Domains > ABC > Log**.
   b) In the administration pane, click the **General** tab, then reviews events for DCN, Replication, Errors, and Device Notifications.

Enabling Domain Monitoring

John enables monitoring on the ABC domain to monitor usage and share key performance indicator data as defined in the service agreement.

Ram does not have access to the monitor information collected; only the Platform administrator can access this functionality and export in a file to share full or part of the monitoring data. Therefore only John can perform this task.

1. In the navigation pane, click **Monitoring**.
2. In the administration pane, click the **General** tab.
3. Click **New** to create a new profile, and type the name `ABCmonitoring`.
4. Select the ABC domain, then check the **Select All Packages of 'ABC'**.
5. Click the **Schedule** tab, then set the default schedule as **Always on**.

6. Enabled the new profile; check **Enable after creation** .
7. Click **OK**.

Deploying the Application Package for Testing

John deploys an application package for testing. During testing, John ascertains which development-cycle properties need to be updated with production-ready values.

John determines the production-ready values by testing the application in a simulated production environment.

1. *Logging in to Sybase Control Center*

   John logs in to Sybase Control Center as the development domain administrator.
2. *Creating and Assigning a Test Security Configuration*

   John creates a new security configuration to authenticate and authorize application users. John creates this security provider to test application-layer security before importing the package into the production environment.
3. *Deploying a Package to the Test Environment*

   Tom provides an application deployment package for replication payloads to John for testing.
4. *Changing Package Properties to Reduce EIS Load*

   John tests the application, then refines package properties based on initial results.
5. *Tuning EIS Connections*

   John configures a connection pool.

See also

- *Setting up a Tenant Domain* on page 17
- *Importing the Package Archive on a Production Server* on page 25

Logging in to Sybase Control Center

John logs in to Sybase Control Center as the development domain administrator.

Because John is part of the domain administrator group that is also mapped correctly in Sybase Control Center, he retains the correct development domain administrator privileges.

1. In Sybase Control Center, enter John's LDAP credentials:
   - **User name** – enter `John`.
   - **Password** – enter `John`.
2. Click **Login**.

Creating and Assigning a Test Security Configuration

John creates a new security configuration to authenticate and authorize application users. John creates this security provider to test application-layer security before importing the package into the production environment.

1. Click **Security** in the navigation pane, and **General** in the administration pane, then click **New**.
2. In **Create Security Configuration**, name the security configuration `AcmeAppSecurity`, then click **OK**.
3. In the navigation pane, expand the **Security** folder and click the new security configuration node.
4. In the administration pane, click the **Authentication** tab then click **New** to add a new LDAP security provider.
5. Select `com.sybase.security.ldap.LDAPLoginModule` as the login module.
6. Configure the LDAP properties.

   The **AuthenticationSearchBase** and **RoleSearchBase** properties ensure that only users and groups in the AcmeCorp organization unit are allowed access to Acme data.

Property	Value
BindDN	`cn=Directory Manager`
BindPassword	`secret`
AuthenticationSearchBase	`ou=AcmeCorp,ou=users,dc=example,dc=com`
DefaultSearchBase	`dc=example,dc=com`
ProviderURL	`ldap://localhost:10389`
RoleMemberAttributes	`uniquemember`
RoleSearchBase	`ou=AcmeCorp,ou=groups,dc=example,dc=com`
AuthenticationScope	`onelevel`
ServerType	`openldap`
RoleScope	`onelevel`

7. Click **OK**.
8. In the **Authentication**, **Authorization**, and **Attribution** tabs, delete **NoSecLoginModule**, **NoSecAuthorizer**, and **NoSecAttributer**, respectively.

9. In the **General** tab, click **Validate**.

   A confirmation message displays upon success.

10. If the validation is successful, click **Apply**.
11. Assign the security configuration to default domain so the security configuration can be tested in the test environment.
    a) Expand the **Domains** folder, then click **Security**.
    b) Click **General** tab, then click **Assign**.
    c) Select **AcmeAppSecurity**.
    d) Click **OK**.

Deploying a Package to the Test Environment

Tom provides an application deployment package for replication payloads to John for testing.

1. In the navigation pane, click **Domains > default > Packages**.
2. In the administration pane, click **Deploy**.
3. In the wizard, browse to `SUP101.jar`, then set the Deployment mode property to **Update**.
4. Click **Next**, then choose default as the deployment domain and choose AcmeAppSecurity as the security configuration to be used for authentication and authorization of the package.
5. Click **Next**, then choose **Map Roles**, so that Tom's logical role of AcmeUser can be mapped to the physical role in the directory (that is, Acme SUP User).
   a) Type a **Role name** of `Acme SUP User`, then click **+**.
   b) Select the newly added role name, and click **Add**.
   c) Click **OK.**
6. Click **Next**, then configure the server connection required for development testing on the default domain.

   Repeat this process for Sales_order and Customer MBOs so that Tom's original MBO endpoint is replaced with the one used for testing purposes.
   a) Select an MBO and choose the sampledb database **Connection**.
   b) Check **Apply connection changes to operations**.
7. Preserve these settings for repeatable deployment to another server generating an XML deployment descriptor:
   a) Create a deployment descriptor.
   b) Click Browse.
   c) Select a directory and type the file name.
   d) Click Save.
8. Click **Finish**.

Changing Package Properties to Reduce EIS Load

John tests the application, then refines package properties based on initial results.

John suggests using DCNs to push updates into the Unwired Server cache in the CDB. This feature helps improve performance and reduce back-end EIS load.

1. Because the device application also uses push notifications, create a subscription template with appropriate settings for these notifications.

   Subscription templates allow all applications that use the template to inherit the same settings.

   a) In the administration pane, click the **Subscriptions** tab, click the **Replication** sub tab, then choose **Templates**.
   b) Click **New**.
   c) Configure these template properties, then click **OK**:

Property	Value
Synchronization group	default
Notification threshold	1 minute
Admin lock	Unlock
Push	Enable

   This configuration enables push notifications to be delivered to device waiting no more than 1 minute since the last synchronization.

2. Update the change detection interval of the package. Change is detected in optimal time, and push notifications are generated for the client to download the changes:

   a) In the administration pane, click the **Synchronization Group** tab, check the group with the **Name** of default, the click **Properties**.
   b) Change the **Change Detection Interval** to 1 minute.

   Note: The change detection interval value needs to be determined very carefully, after considering application requirements of data consistency and data concurrency. Change detection triggers a diff-calculation for sending data changes to subscribed clients. These calculations can adversely affect performance. Therefore, the administrator and developer must jointly determine this interval value before configuring it in a production environment.

Tuning EIS Connections

John configures a connection pool.

1. In the navigation pane, expand the **default** domain, then click **Connections**.
2. In the administration pane, select **sampledb**, then click **Properties**.

3. Add the Max pool size property, and change its value to `100`.
4. Click **Test Connection** and make sure the server can still be pinged.
5. If you can connect to the server, click **Save**.

Importing the Package Archive on a Production Server

John exports the package from the default domain, then imports the package on a production Unwired Server to deploy it without needing to reconfigure the package.

Prerequisites

Export the package that was deployed in the previous step.

1. Change the security configuration of the package that was deployed in the previous step.
   a. In the navigation pane, expand **Domains**, **default**, then click **Security**.
   b. In the administration pane, click **Assign**.
   c. Select ABCAppSecurity and click **OK**.
   d. In the navigation pane, expand **Domains**, **default**, **Packages**, then click the package (`SUP101:1.0`).
   e. In the administration pane, click the **Settings** tab.
   f. Change the security configuration to ABCAppSecurity.
   g. Click **Save**.
2. Export the package that was deployed in the previous step.
   a. Under **Domains**, expand **default**, then select **Packages**.
   b. Click the package (`SUP101:1.0`), then click **Export**.
   c. In the Export Package dialog, click **Next**.
   d. Click **Finish**.

Before importing the package archive, create the sampledb connection profile in the ABC domain.

1. In the navigation pane, expand the ABC domain, then click **Connections**.
2. In the administration pane, select the **Connections** tab, then click **New**.
3. Type the Connection pool name `sampledb`, select the JDBC Connection pool type, then select **Sybase ASA template**.
4. Set the following properties for the sampledb connection pool:

```
Commit Protocol : optimistic
Server Name : localhost
Service Name : sampledb
Port Number : 5500
User : dba
Password : sql
Max Pool Size : 100
```

5. Click **Test Connection** and make sure there is a valid connection to the server.
6. Click **OK**.

Task

In this step, John uses the same system. However you should consider this task as being typically performed on a separate production system.

1. Under **Domains**, expand **ABC** > **Packages**.
2. In the administration pane, click **Import**, then select the ZIP archive (`SUP101:1.0.zip`) and click **Import**.
   The **Import** dialog shows a displays a message that the server connections with the same name (sampledb) will be overwritten.
3. Review the package settings, including the pre-configured security configuration (and role mappings), log level, cache group, sync group, role mapping, subscription templates, and connection settings. Those properties are re-instated exactly as configured at export time.

Activity 3: Automate Application Connection Registration

Goal: Automate the registration of application connections, which represent a client application on a device created for the device user.

Having previously tested this functionality, John knows that he can use Sybase Control Center to register application connections. He can also review application users that have been successfully authenticated.

Acme anticipates the number of device users to grow exponentially over time. Therefore, John wants to automate device registration. John knows that Unwired Platform has public APIs for customizing and automating device and administration applications. Further reading leads John to discover that he can use the Management API to build an application that batches application connection registrations, which streamlines processes considerably.

This exercise is an introduction to the Management APIs, which provide a programmatic interface to the functionality available in the Sybase Control Center administrative console. For complete details, see *Developer Guide: Unwired Server Runtime > Management API*.

Business Requirements for Synchronization

Because Acme develops and hosts synchronization applications for their customers, requirements for this activity revolve around streamlining processes and maintenance.

Requirement	Action Required
As their client base grows, Acme must have a predictable, stable, and scalable methodology for registering new application connections. Registering one application connection at a time is not viable with only one Unwired Platform administrator.	Investigate the Unwired Server Management API to determine whether registrations can be performed in batches.

Technical Prerequisites

Prepare to batch application connection registrations by completing these prerequisites.

This activity requires a developer to create a new project in Sybase Unwired Workspace, and add the JAR files for the Administration API to the project. Sybase provides the project and Java code in the `SUP101_AdminAPI.zip` file on Sybase Product Documentation. Then you can import the project in to Unwired WorkSpace.

- If you are viewing this guide online from the Sybase Product Documention Web site, click *SUP101_AdminAPI.zip* to access the zip archive containing the Unwired WorkSpace project file and Java code file.
- If you are viewing this guide as a PDF, go to the Sybase Product Documentation Web site at *http://sybooks.sybase.com/nav/summary.do?prod=1289&lang=en&submit=%A0Go%A0&prodName=Sybase+Unwired+Platform&archive=0*. Click the link for the Sybase Unwired Platform version that you want. Then, navigate to this topic in the tutorial, and click the link for the zip file to download it.

Writing the Java Code for Batch Registration

John or his Java-savvy developer reviews the Developer Guide for Unwired Server Runtime > Management API and decides to write Java code to automate application connection registration outside of Sybase Control Center.

1. Use Unwired Workspace to open the `RegisterApplicationConnections.java` class from the `src` folder.
2. Update these properties based on what you indicated during installation.

```
public class RegisterApplicationConnections {
```

```
    private String supHost = "localhost";
    private int supPort = 2001;
    private String supAgentHost = "localhost";
    private int supAgentPort = 9999;
    private String supAdministrator = "supAdmin";
    private String supAdministratorPassword = "AdminPwd";
```

3. Set the application ID and domain for the application connection by inserting the following code in the public `AppConnectionSettingVO getSettings()` method:

```
setting.put(APPCONNECTION_SETTING_FIELD.APPLICATION_ID,
"SUP101");
setting.put(APPCONNECTION_SETTING_FIELD.DOMAIN, "ABC");
```

4. Scan the Java file to determine if any other parameters should change based on your environment.

   If a Relay Server farm with an RSOE is set up and configured, indicate the Relay Server host, Relay Server port number, and farm ID (indicated by zero in the example).

5. Save the changes.

Validating the Automated Registration

John tests the functionality of the modified code using both Unwired WorkSpace and Sybase Control Center.

1. In Unwired WorkSpace:
   a) In the main toolbar, click the green arrow.
   b) In the console view, check that the output message lists the newly registered device, for example:
   `Application connection 10 registered for user1`
2. In Sybase Control Center:
   a) In the navigation pane, click **Applications**, then select the **Application Connections** tab.
   b) The application connection that appeared in Unwired WorkSpace is also registered in this tab.
   c) Select this application connection and click **Properties** to see the inherited settings from the application connection template used for the registration.
   d) On the **Application Connections** tab, validate the pairing of the application connection and the user specified in the registration code.

   Note: Only initial registration values appear with automated registration. Other values change once the user activates the device.

Learn More About Sybase Unwired Platform

Once you have finished, try some of the other samples or tutorials, or refer to other development documents in the Sybase Unwired Platform documentation set.

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

Tutorials

Try out some of the other getting started tutorials available on the Product Documentation Web site to get a broad view of the development tools available to you.

Example Projects

An example project contains source code for its associated tutorial. It does not contain the completed tutorial project. Download example projects from the SAP® Community Network (SCN) at *http://scn.sap.com/docs/DOC-8803*.

Samples

Sample applications are fully developed, working applications that demonstrate the features and capabilities of Sybase Unwired Platform.

Check the SAP® Development Network (SDN) Web site regularly for new and updated samples: *https://cw.sdn.sap.com/cw/groups/sup-apps*.

Online Help

See the online help that is installed with the product, or available from the Product Documentation Web site.

Developer Guides

Learn best practices for architecting and building device applications:

- *Mobile Data Models: Using Data Orchestration Engine* – provides information about using Sybase Unwired Platform features to create DOE-based applications.
- *Mobile Data Models: Using Mobile Business Objects* – provides information about developing mobile business objects (MBOs) to fully maximize their potential.

Use the appropriate API to create device applications:

- *Developer Guide: Android Object API Applications*
- *Developer Guide: BlackBerry Object API Applications*
- *Developer Guide: iOS Object API Applications*
- *Developer Guide: Windows and Windows Mobile Object API Applications*

- *Developer Guide: Hybrid Apps*

Customize and automate:

- *Developer Guide: Unwired Server Runtime > Management API* – customize and automate system administration features.

Javadoc and HeaderDoc are also available in the installation directory.

Index

O

P

R

S

T

U

W