SYBASE®
An SAP® Company

Administration Workbook

Sybase Unwired Platform 2.0

DOCUMENT ID: DC01625-01-0200-01
LAST REVISED: April 2011

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 aHighly Available and Secure Development Environment	Create a single-node relay server-enabled cluster for a development/test environment.	Read: • (Recommended) *System Administration of the Unwired Platform > Systems Design* • (Recommended) *System Administration of the Unwired Platform > Security Administration*

Workbook activity	Goal	Co-requisites
Activity 2: Setting Up Multiple Domains	Set up multiple domains and use them to support different tenants.	Complete: • (Required) *Activity: Setting Up a Highly Available and Secure Environment* Read: • (Recommended) *System Administration of the Unwired Platform > Systems Design > Multitenant Environments* • (Recommended) *System Administration of the Unwired Platform > System Administration > Server Environment Administration > Domain Administration Overview*

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 their 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 on a specific actor. This perspective allows you to understand how tasks interrelate in a given deployment of Unwired Platform.

The environment:	Acme Corporation purchase 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. They will 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.

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 will self-delegate this role to his existing user profile. • SUP Developer – John sets Tom up for this role, so that Tom can deploy applications and use the system for testing. • SUP User – John sets Mary up for with this role, so she can test Tom's application.
Security policy requires all default passwords be changed.	John needs to change the CDB password used by Unwired Platform by default to one that is unique for Acme.

Requirement	Actions required
ABC Corporation is a client that requires hosted mobility services. Domain-level security must be configured accordingly.	John must ensure that Ram be set up as domain administrator of ABC's domain and artifacts, and a security configuration must be created 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 2-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 installing relay server and configuring relay server outbound enablers (RSOEs) to use this service during access requests.
Cache updates must be via 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, *Sybase Unwired Platform Install Guide > Installing Developer Editions > Performing a Cluster Installation of Developer Edition*.
- Deploy relay server on an IIS 7.x host. For details, see *System Administration > Environment Setup > Relay Server Setup > Installing the Relay Server Components to IIS 7.x on Windows*.
- Use the OpenDS LDAP server installed with development editions of Unwired Platform. To complete this task flow, the LDAP service must running.
- Download Apache DS Studio from *http://directory.apache.org/studio/*.

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.

1. *Securing the Administration Infrastructure*

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

2. *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.

3. *Preparing for Connections Outside the Firewall*

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

4. *Securing the DCN Transport*

   John needs to set up and test the DCN port. Because the imported private key used for DCN encryption needs to be an X.509 certificate, John uses the Java keytool utility to create a self-signed certificate.

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.

3. *Configuring Sybase Control Center Security*

   Once John logs shuts down Sybase Control Center, he must modify the configuration file for Sybase Control Center, so that it can also authenticate against the same LDAP directory as Unwired Server.

4. *Removing Preconfigured Platform Users*

   Once John has completed the security configuration, he notifies Jane who then removes preconfigured users from the OpenDS repository.

5. *Changing the CDB Password and Registering Changes Among Components*

   The Platform administrator can update the modified password in Unwired Server configuration files.

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. *Configuring OpenDS Import Encoded Passwords From LDIF*

   Because Acme has an LDIF file that includes encoded passwords, Jane must configure OpenDS 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.

Configuring OpenDS Import Encoded Passwords From LDIF

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

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

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

Page	Property	Value
Authentication		
	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**.

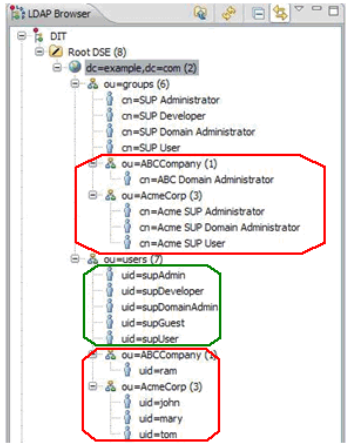
Importing LDIF Contents

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 each 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 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 default Platform administration login credentials.

1. *Logging in as Platform Administrator*

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

   John modifies the "admin" security configuration to use the OpenDS 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.

Logging in as Platform Administrator

John logs in to Sybase Control Center with the preconfigured login credentials to access Unwired Server.

1. In Sybase Control Center, enter the default credentials:
   - **User name** – enter the default user name of supAdmin.
   - **Password** – enter the default supAdmin password of s3pAdmin.
2. Click **Login**.

John sees the full view of Unwired Platform.

Configuring the Admin Security Configuration

John modifies the "admin" security configuration to use the OpenDS 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. Modify the LDAP provider to change the default authentication and role scope (the original values use the entire tree). Use these new values:

Property	New Value
AuthenticationScope	subtree
RoleScope	subtree

   For information about the complete set of configuration properties available, see *System Administration > System Reference > Security Provider Configuration Properties > LDAP Configuration Properties*.
4. Click **OK**.

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 folder, then click the security configuration named **admin**.
2. Click **Map Role**.
3. Map these roles:

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

   For details, see *System Administration > Security Administration > Security Layers > User Security Setup > Security for Administration Users > Mapping and Assigning Unwired Platform Default Roles*.

4. Remove the default mappings for SUP Administrator and SUP Domain Administrator.

   This removes the default login access of supAdmin/s3pAdmin and supDomainAdmin/s3pDomainAdmin logins. Once unmapped, no user can use these credentials to log in to Sybase Control Center.
5. Click **OK**.
6. Log out from Sybase Control Center.

Configuring Sybase Control Center Security

Once John logs shuts down Sybase Control Center, he must modify the configuration file for Sybase Control Center, so that it can also authenticate against the same LDAP directory as Unwired Server.

This task effects changes against two files:

1. Use a text editor to open `<UnwiredPlatform_InstallDir>\SCC-XX\conf\csi.properties`.
2. Add lines that define the 'subtree' level as the authentication and authorization scope for the Unwired Platform LDAP login module:

   ```
   CSI.loginModule.XX.options.RoleScope=subtree
   CSI.loginModule.XX.options.AuthenticationScope=subtree
   ```
3. Save the changes and close the file.
4. Use a text editor to open `<UnwiredPlatform_InstallDir>\SCC-XX\conf\roles-map.xml`.
5. Add these lines for the same Unwired Platform LDAP login module:

   ```
   <role-mapping modRole="Acme SUP Administrator"
   uafRole="uaAnonymous,uaAgentAdmin,uaPluginAdmin,sccAdminRole,sccU
   serRole" />
   <role-mapping modRole="Acme SUP Domain Administrator"
   uafRole="uaAnonymous,uaAgentAdmin,uaPluginAdmin,sccUserRole" />
   <role-mapping modRole="ABC Domain Administrator"
   uafRole="uaAnonymous,uaAgentAdmin,uaPluginAdmin,sccUserRole" />
   ```

   These lines map logical Sybase Control Center roles to the LDAP directory physical roles. Specifically, the first line mapping for sccAdminRole gives 'Acme SUP Administrator' access to Sybase Control Center as administrator.

   Note: The Sybase Control Center infrastructure can be separately secured and managed by an administrator who may be a different from the administrator for Unwired Platform. In this scenario, John is administrator of both Sybase Control Center and Unwired Platform infrastructures.

   As a Sybase Control Center administrator, the users granted this role can perform administration and configuration tasks from the Unwired Platform management console after a successful login.

The last 2 entries give ‘Acme Domain Administrator’ and ‘ABC Domain Administrator’ access as a SCC User (which is mapped sccUserRole).

6. Save the changes and close the file.
7. Restart the Sybase Unified Agent service.

Removing Preconfigured Platform Users

Once John has completed the security configuration, he notifies Jane who then removes preconfigured users from the OpenDS repository.

Preconfigured users should be removed from active deployments of Unwired Platform.

1. Launch Apache DS Studio.
2. Browse the Unwired Platform LDAP repository and delete these entries:
   - supAdmin
   - supDomainAdmin
   - supDeveloper
   - supUser

Changing the CDB Password and Registering Changes Among Components

The Platform administrator can update the modified password in Unwired Server configuration files.

1. Open `<UnwiredPlatform_InstallDir>\Sybase\UnwiredPlatform\Servers\UnwiredServer\Repository\Instance\com\sybase\sup\server\SUPServer\sup.properties`.
2. Modify the cdb.password and cldb.password properties to set a new password.

   Passwords are entered in clear text password. After you have completed the next step, all passwords are encrypted and the sup.properties file is resaved with the new encrypted values.
3. Run `<UnwiredPlatform_InstallDir>\Sybase\UnwiredPlatform\Servers\UnwiredServer\bin\configure-mms.bat <clustername>`.

   In a single server setup, <clustername> is host name of this computer. In a cluster setup, <clustername> is the hostname of first node installed in the cluster.

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.

Because Jane removed the default Platform administration login credentials from LDAP, John knows he must now log in with his own login credentials as defined in his LDAP user account. 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.
5. *Removing and Reinstalling Services*

   To apply changes, John removes existing services and reinstalls them.

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, click **Performance Configuration**.
4. Change the **Minimum Heap Size** and the **Maximum Heap Size** properties to `512M`.
5. 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 **Server Configuration** view, click the **Replication** tab.
2. Change the **RBS Synchronization Cache Size** property to `80M` and **Thread Count** property to `50M`.
3. Click **Save**.. If there are no errors, the changes are written to the configuration file. 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.

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.

Removing and Reinstalling Services

To apply changes, John removes existing services and reinstalls them.

In the Acme development and test, environment the CDB is installed on the same node as Unwired Server.

1. Stop Unwired Platform services:
   a) In Sybase Control Center, expand the Servers folder in the navigation pane, select the primary server node, then in the administration pane click **Stop**.
   b) In the Windows Administration Tools Services dialog, locate the CDB service and stop it.
2. Remove Unwired Platform services by running these commands at a command prompt:

```
cd c:\Sybase\UnwiredPlatform\Servers\UnwiredServer\bin
sup-server-service remove
```

The CDB service state changes to disabled.

3. To correct the CDB service state, remove the CDB service by running this command at a command prompt:

   ```
   sc delete SybaseUnwiredPlatform<hostname>Database1
   ```

4. Restart the computer.
5. Install Unwired Platform services by running these commands at a command prompt:

   ```
   cd c:\Sybase\UnwiredPlatform\Servers\UnwiredServer\bin
   sup-server-service install auto -sampledb
   ```

   In Acme's environment, they use the auto option to automatically start the services. They also use the -sampledb option, so that the sampledb database is started as a service with the server and available to their developers.
6. In the Window Administration Tools Services dialog, ensure all SUP Services (those with a `SybaseUnwiredPlatform*` prefix) are started.

Preparing for Connections Outside the Firewall

Relay server allows Mary to test RBS device clients in a development test environment. Mary will test device clients 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 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 versions of Unwired Servers.
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. In the administration pane, click the Relay Server 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	RBS

7. Click the + button.
8. Define the MBS farm:

Property	Value
Farm ID	SUPMBSFarm
Type	MBS

9. Click the + button.
10. Click **Next**.
11. Assign the RBS Farm to a node so that devices with RBS request can connect to the relay server.
    a) Select SUPRBSFarm:
    b) Configure these properties:
       - **Node ID –** PrimarySUPDev
       - **Token –** abcd
    c) Click the + button.

12. Click **Finish**.
13. Generate the configuration to transfer the properties from the cluster database to the relay server host machine.
    a) Click **Generate**.
    b) Choose **Relay server properties configuration file** and output the whole file by selecting **Whole Relay Server**.
    c) Click **Finish** and transfer the generated file to the corresponding host computer.

Setting Up the RSOE for Unwired Server RBS Services

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

Because connections will be secured using HTTPS, he must also load the correct certificate file.

1. In the navigation pane, click **Servers** > <**ServerNode**> > **Server Configuration**.
2. Upload the certificate:
    a) In the administration pane, select the **RSOE** tab, then click **Certificate Files**.
    b) Click the + button. Browse and select the `.CRT` file to upload, then click **Open**.
    c) Click **OK**.
3. Configure the RSOEs. While Sybase recommends three RSOEs per synchronization type; however for a development environment, one should be sufficient.
    a) In the administration pane, select the **RSOE** tab, then click **New**.
    b) Configure general properties:

Property	Value
Farm type	RBS
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
TLS type	RSA
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 **RSOE** 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 Reference* that corresponds to the device platform.

Securing the DCN Transport

John needs to set up and test the DCN port. Because the imported private key used for DCN encryption needs to be an X.509 certificate, John uses the Java **keytool** utility to create a self-signed certificate.

1. *Creating Self-Signed Encryption Certificate*

   John creates a self-signed certificate with the Java SDK keytool utility before he imports it into the Unwired Platform for DCN encryption.
2. *Setting Up a DCN Security Profile with the Self-Signed Certificate*

   John uses Sybase Control Center to finalize the DCN security setup by importing the keytool keys and using values defined with that utility.
3. *Enabling the DCN Security Profile*

   John also uses Sybase Control Center to enable the new DCN security profile.

Creating Self-Signed Encryption Certificate

John creates a self-signed certificate with the Java SDK **keytool** utility before he imports it into the Unwired Platform for DCN encryption.

Prerequisites

To use the **keytool** utility, John knows he must set the `JAVA_HOME` environment variable to the JDK directory used by Unwired Platform, in addition to defining `%JAVA_HOME%\bin` as a the path variable. to the path variable, because keytool utility needs this setting.In this scenario, John uses the default of `C:\Sybase\UnwiredPlatform\JDK1.6.0_24`.

Task

Note: For details about all supported utility options beyond those illustrated below, see *System Administration > System Reference > Command Line Utilities > Certificate and Key Management Utilities > Key Tool (keytool) Utility*.

1. Change directory to `<UnwiredPlatform_InstallDir>\Repository\Security`, and run this command:

```
keytool -genkey -alias dcn -keypass changeit -keyalg RSA -keysize
1024 -validity 3650 -keystore dcn.jks
-storepass chageit
```

Follow the prompts to generate a public-private keypair for the Acme organization.

Prompt	Value Entered
Name	`John`
Organization	`Acme Corporation`
Organizational Unit	`Development`
City	`Dublin`
State/Province	`CA`
Country Code	`US`

2. Use **keytool** to import the keystore to the destination keystore by using this command:

```
keytool -importkeystore -destkeystore keystore.jks -deststorepass
changeit -srckeystore dcn.jks
-srckeypass changeit -alias DCN
```

3. Use **keytool** to export just the public key to the local disk.

```
keytool -keystore keystore.jks -storepass changeit -alias DCN -
export -file C:\temp\dcn.crt
```

Setting Up a DCN Security Profile with the Self-Signed Certificate

John uses Sybase Control Center to finalize the DCN security setup by importing the **keytool** keys and using values defined with that utility.

For more information, see *Sybase Control Center > Configure > Configuring Unwired Platform > Unwired Server > Server Properties > General Server Ports > Configuring SSL Properties > Creating an SSL Security Profile*.

1. In the navigation pane, expand the **Servers** folder, then click **Server Configuration**.
2. In the administration pane, click the **General** tab, then select **SSL Configuration**.
3. Create a new security profile by entering these values in an empty row of the table.

   These alias value is the same value set with the **keytool** -alias property in the previous task.

Column Name	Value
Security Profile	`dcn_profile`
Certificate Alias	`dcn`
Authentication	`intl`

4. Click **Save**.

5. Click **Restart** to apply pending changes.

Enabling the DCN Security Profile

John also uses Sybase Control Center to enable the new DCN security profile.

1. On the General tab, click the **Communication Ports**.
2. In the **SSL Profile** column for the port number 8001, select **Enabled** then choose **dcn_profile**.
3. Restart the Unwired Server to enable the HTTPs DCN settings.
4. Deploy a package that contains an application that uses DCN and test the DCN settings with it.

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 their Unwired Platform domain.

Requirement	Action required
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 their users if an update is 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:

- Complete *Activity: Setting Up an Available and Secure Development Environment* on page 3.
- Ensure Unwired Server and Sybase Unified Agent 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 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.

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 Configuration** in the navigation pane, then in the administration pane, click **New**.
3. In **Create Security Configuration**, name the security configuration `ABCAppSecurity`, then click **OK**.

4. In the navigation pane, expand the **Security Configuration** 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. Configure the LDAP properties as follows:

   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	`one-level`
ServerType	`openldap`
RoleScope	`one-level`

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

   A confirmation message is displayed upon success.
10. 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. Select **ABCAppSecurity** as the default security provider for the ABC domain.
5. 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 either of these users log into Sybase Control Center, they must be identified as Domain administrator. Ram was already defined as a member “ABC Domain Administrator” role when the LDIF file was imported into OpenDS. In addition, John determines that Tom needs to access the default domain. Tom is also already a member of “Acme SUP Domain Administrator” group in OpenDS.

Note: Mapping roles to allow administrator access in SCC was configured in Activity 1 of this work book.

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, 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, **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, 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, click the default domain **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** > **Logs**.
   b) In the administration pane, click the **Settings** tab.
   c) Click **Enable** and set data retention to `10` days.
   d) Click **Save**.
2. Review package-level logging events:
   a) In the navigation pane, click **Domains** > **ABC** > **Logs**.
   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'** box.
5. Click the **Schedule** tab, then set the default schedule as **Always on**.
6. 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 in order to test application-layer security before importing the package into the production environment.

3. *Deploying a Package to the Test Environment*

   Tom provides an RBS application deployment package 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.

6. *Deploying the Tested Package to a Production Server*

   As testing is completed and production-ready properties are set, John takes a copy of the package and deploys it to a production runtime.

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 which is also mapped correctly in Sybase Control, 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 in order to test application-layer security before importing the package into the production environment.

1. Click **Security Configuration** in the navigation pane, then in the administration pane, click **New**.
2. In **Create Security Configuration**, name the security configuration `AcmeAppSecurity`, then click **OK**.
3. In the navigation pane, expand the **Security Configuration** 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. Configure the LDAP properties as follows:

   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	`one-level`
ServerType	`openldap`
RoleScope	`one-level`

6. Click **Save**.
7. In the **Authentication**, **Authorization**, and **Attribution** tabs, delete **NoSecLoginModule**, **NoSecAuthorizer**, and **NoSecAttributer** respectively.
8. In the **General** tab, click **Validate**.

   A confirmation message is displayed upon success.
9. If the validation is successful, click **Apply**.
10. Assign the package to default domain so the package and the security configuration can be tested in the test environment.
    a) Expand the **Domains** folder, then click **default**.
    b) Click **Security Configurations** tab, then click **Assign**.
    c) Select AcmeAppSecurity.

Deploying a Package to the Test Environment

Tom provides an RBS application deployment package to John for testing.

1. In the navigation pane, click **Domains** > **default**.
2. In the administration pane, click **Deploy**.
3. In the wizard, browse to `SUP101.jar`, then configure these properties:

Property	Value
Synchronization mode	Replication

Property	Value
Deployment mode	Replace

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 OpenDS 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.
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. In the navigation pane, click **Domains > default > Packages > SUP101:1.0**.
2. Disable cache groups, so that the Unwired Server cache is only by DCN requests:
   a) In the administration pane, click the **Cache Group** tab.
   b) Select the **Default** cache-group, then click **Properties**.
   c) Check the NEVER expiry option, then click **OK**.
3. Since the RBS 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, 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.

4. Update the change detection interval of the package, so that push notifications are sent if a change is detected:
   a) In the administration pane, click the **Synchronization Group** tab, check the group with a **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. Change the **Max pool size** property 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**.

Deploying the Tested Package to a Production Server

As testing is completed and production-ready properties are set, John takes a copy of the package and deploys it to a production runtime.

1. In the navigation pane, expand **default** > **Packages**.
2. In the administration pane, select the package pre-configured for the production system (SUP101:1.0), then click **Deploy**.
3. Follow the instructions of the wizard and click **Finish**.

Importing the Package Archive on a Production Server

John imports the package on a production Unwired Server to deploy it without needing to reconfigure the package.

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

1. In the navigation pane, expand **ABC** > **Packages**.
2. In the administration pane, click **Import**, then select the ZIP archive (`SUP101:1.0.zip`) and click **OK**.
   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.

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://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.infocenter.pubs.docset-SUP-2.0.0/doc/html/title.html*.

Tutorials

Try out some of the other getting started tutorials to get a broad view of the development tools available to you.

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 updates: *http://www.sdn.sap.com/irj/sdn/mobile*. Click on Sybase Unwired Platform and navigate to Samples.

Online Help

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

Developer Guides

Learn about using the API to create device applications:

- *Developer Guide for BlackBerry*
- *Developer Guide for iOS*
- *Developer Guide for Mobile Workflow Packages*
- *Developer Guide for Windows and Windows Mobile*

Customize and automate:

- *Developer Guide for Unwired Server Management API* – customize and automate system administration features.
- *Developer Guide for Unwired Server* – customize and automate server-side implementations for device applications, and administration, such as data handling.

Javadoc and HeaderDoc are also available in the installation directory.

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

W