SYBASE®
An SAP® Company

Release Bulletin 2.1 ESD #3

Sybase Unwired Platform 2.1 ESD #3

DOCUMENT ID: DC00835-01-0213-17
LAST REVISED: January 2014

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

Contents

Release Bulletin 2.1 ESD #3 Patch 02 (SUP 2.1 SP03 PL02)

The 2.1 ESD #3 Patch 02 (SUP2.1 SP03 PL02) provides fixes and updates for Sybase® Unwired Platform 2.1 ESD #3, and 2.1 ESD #3 Patch 01.

Note:

- Sybase Unwired Platform 2.1 ESD #3 Patch 02 is a cumulative patch which must be installed on a system with Sybase Unwired Platform 2.1 ESD #3 already installed. You do not need to upgrade to 2.1 ESD #3 Patch 01 prior to installing this patch.
- If you have already installed Unwired Platform Runtime 2.1 ESD #3 Patch 01, you do not need to patch to Patch 02 for Runtime. However, if you choose to install Runtime Patch 02, it will not compromise the runtime environment.

For fixes to known issues, see *Fixed Issues in Patch 02* on page 11. For patch contents, and a complete list of fixes, see the *README* for Sybase Unwired Platform 2.1 ESD #3 Patch 02.

Patch Readiness Checklist for Unwired Platform Runtime 2.1 ESD #3 Patch 02

Ensure your environment is ready for patching to Unwired Platform Runtime 2.1 ESD #3 Patch 02 or 2.1 ESD #3 Patch 01.

1. Verify that Unwired Platform installations are at version 2.1 ESD #3, or 2.1 ESD #3 Patch 01.
2. Back up each node in your system infrastructure.
3. Back up the registry on each node in your system infrastructure.
4. Back up the shared database if you have Unwired Platform installed in a cluster. For example, back up the cache, cluster, monitor, and domainlog databases, and if present the sample database.
5. Disable automatic system backups for the 2.1.3 SDK installation tree. If you have Unwired Platform installed in a cluster, ensure that automatic system backups are disabled before upgrading each node in the cluster.
6. Disable virus scan on the installation tree. If you have Unwired Platform installed in a cluster, disable the virus scan on each node of the cluster. Alternatively, disable virus scan altogether while the patch is being applied.
7. Make sure that the available disk space on each data tier node in a cluster, or on a single-server installation, is sufficient to accommodate a 50% increase in the size of the shared database.

8. For virtual machine (VM) infrastructure deployments, take a snapshot of each node before proceeding.
9. Adjust User Access Control for your operating system, then validate that the environment continues to work as expected:
   - On Windows 7 or Windows 2008 R2, change the UAC setting for Windows users to **Never Notify** and restart Windows.
   - On Windows Vista and Windows 2008, disable UAC, and restart Windows.
10. Verify that you have administrator privileges on the patch target hosts.
11. Verify that you are using the same Windows user account that was previously used to install Unwired Platform.
12. Verify that database administrator user names and passwords for the configurable Unwired Platform databases have not changed from the default settings ("dba"/"sql"), or enter the new values here:

Database	Admin User Name	Password
Sybase Unwired CacheDB		
Sybase Unwired ClusterDB		
Sybase Unwired LogDataDB		
Sybase Unwired MessagingDB		

Patching to Unwired Platform Runtime 2.1 ESD #3 Patch 02

Use Sybase Unwired Platform 2.1 ESD #3 Patch 02 to patch the Runtime environment in Unwired Platform 2.1 ESD #3. If you are using an earlier version of Unwired Platform, you must upgrade to 2.1 ESD #3 before applying the patch.

Note: If you have already installed Unwired Platform Runtime 2.1 ESD #3 Patch 01, you do not need to patch to Patch 02. However, if you choose to install Patch 02, it will not compromise the runtime environment.

Tip: While the patch script verifies environment readiness, use *Patch Readiness Checklist for Unwired Platform Runtime 2.1 ESD #3 Patch 02* to avoid errors and interruptions in the patch process.

Patching a Single Node

In a single-node installation, all Unwired Platform server components are located on a single server.

1. Extract the patch archive file to a temporary location, for example, `C:\tmp\2.1.3p2`.

2. Log in with administrator privileges, then open a Windows Command Prompt (cmd.exe) and change to the folder where you extracted the patch files.

   On Windows 7 and Windows Server 2008R2, start the Command Prompt with the Run as Administrator option.
3. Run `installebf.bat`.
   - If the script detects environment issues, you see an error message and the patch process stops. Follow the instructions provided and restart the patch. If you cannot correct the issue, contact Sybase Technical Support.
   - If the patch script stops with an unexpected error message, look for locked files in the installation directory tree. Unlock any locked files and re-run the patch script.

   Note: The script cannot specifically detect locked files in the installation directory tree. The error message that appears when the script encounters locked files is unpredictable.
4. Press **Y**, then **Enter** to install the patch.

Patching a Simple Cluster

A simple cluster consists of one data tier node and one or more Unwired Server nodes, with failover capability between multiple Unwired Server nodes.

1. Shut down the nodes in this order:
   a. Secondary Unwired Servers – shut down services in this order:
      1. Sybase Unwired Server
      2. Sybase Control Center
      3. Sybase Unwired SampleDB (not applicable to Enterprise Server Edition)
   b. Primary Unwired Server – shut down the same services in the same order as for Secondary Unwired Servers.
   c. Data tier server – shut down these services, in any order:
      1. Sybase Unwired CacheDB
      2. Sybase Unwired Cluster DB
      3. Sybase Unwired LogDataDB
2. Install the patch on each node in the cluster, in this sequence:
   - Data tier node
   - Primary Unwired Server
   - Secondary Unwired Servers

   Note: Do not begin patching any secondary servers until the primary server has completely restarted all its services after being patched.

   a) Extract the patch archive file to a temporary location, for example, `C:\tmp\2.1.3p2`.
   b) Log in with administrator privileges, then open a Windows Command Prompt (cmd.exe) and change to the folder where you extracted the patch files.

On Windows 7 and Windows Server 2008R2, start the Command Prompt with the Run as Administrator option.

c) Run `installebf.bat`.

- If the script detects environment issues, you see an error message and the patch process stops. Follow the instructions provided and restart the patch. If you cannot correct the issue, contact Sybase Technical Support.
- If the patch script stops with an unexpected error message, look for locked files in the installation directory tree. Unlock any locked files and re-run the patch script.

Note: The script cannot specifically detect locked files in the installation directory tree. The error message that appears when the script encounters locked files is unpredictable.

d) Press **Y**, then **Enter** to install the patch.

e) When the patch finishes, review the text on your screen for error messages.

On successful completion of the patch, the services for the patched server are started, even if you have specified to manually start services.

Note: After patching the data tier, the first Unwired Server may take as long as an hour to restart.

f) Before proceeding, verify that the patched server is started.

Patching a Microsoft Cluster

A Microsoft cluster consists of two data tier nodes and multiple Unwired Server nodes, with failover between data tier nodes managed by Microsoft Cluster Service, also called Failover Clustering.

1. Shut down the nodes in this order:
   a. Secondary Unwired Servers – shut down these services in this order:
      1. Sybase Unwired Server
      2. Sybase Control Center
      3. Sybase Unwired SampleDB (not applicable to Enterprise Server Edition)
   b. Primary Unwired Servers – shut down the same services in the same order as for the Secondary Unwired Servers.
2. On the active data tier node, launch the MS Cluster Administrator and make sure the following active Unwired Platform resources are offline:
   - Afaria® Database Service (available with Unwired Platform version 1.5.2 or earlier)
   - Sybase Unwired CacheDB service
   - Sybase Unwired ClusterDB service (available with Unwired Platform version 2.0 or later new installations)
   - Sybase Unwired LogDataDB service (available with Unwired Platform version 2.0 or later new installations)

3. Install the patch on the active data tier node:
   a) Extract the patch archive file to a temporary location, for example, `C:\tmp\2.1.3p2`.
   b) Log in with administrator privileges, then open a Windows Command Prompt (cmd.exe) and change to the folder where you extracted the patch files.

      On Windows 7 and Windows Server 2008R2, start the Command Prompt with the Run as Administrator option.
   c) Run `installebf.bat`.
      - If the script detects environment issues, you see an error message and the patch process stops. Follow the instructions provided and restart the patch. If you cannot correct the issue, contact Sybase Technical Support.
      - If the patch script stops with an unexpected error message, look for locked files in the installation directory tree. Unlock any locked files and re-run the patch script.

      Note: The script cannot specifically detect locked files in the installation directory tree. The error message that appears when the script encounters locked files is unpredictable.
   d) Press **Y**, then **Enter** to install the patch.
   e) When the patch finishes, review the text on your screen for error messages.

      Note: After patching the data tier, the first Unwired Server may take as long as an hour to restart.
   f) Before proceeding, verify that the patched server is started.
4. Make the passive node active and ensure all database services are offline.
   a) Launch Microsoft Cluster Administrator.
   b) Fail over the active node to the passive node.

      The node that was passive is now active.
5. Install the patch on the new active data tier node:
   a) Extract the patch archive file to a temporary location, for example, `C:\tmp\2.1.3p2`.
   b) Log in with administrator privileges, then open a Windows Command Prompt (cmd.exe) and change to the folder where you extracted the patch files.

      On Windows 7 and Windows Server 2008R2, start the Command Prompt with the Run as Administrator option.
   c) Run `installebf.bat`.
      - If the script detects environment issues, you see an error message and the patch process stops. Follow the instructions provided and restart the patch. If you cannot correct the issue, contact Sybase Technical Support.
      - If the patch script stops with an unexpected error message, look for locked files in the installation directory tree. Unlock any locked files and re-run the patch script.

Note: The script cannot specifically detect locked files in the installation directory tree. The error message that appears when the script encounters locked files is unpredictable.

d) Press **Y**, then **Enter** to install the patch.

e) When the patch finishes, review the text on your screen for error messages.

Note: After patching the data tier, the first Unwired Server may take as long as an hour to restart.

f) Before proceeding, verify that the patched server is started.

6. Install the patch on the Unwired Server nodes, in this sequence:

- Primary Unwired Server – shut down services in this order:
  a. Sybase Unwired Server
  b. Sybase Control Center
  c. Sybase Unwired SampleDB (not applicable to Enterprise Server Edition)
- Secondary Unwired Servers – shut down same services in same order as for the primary Unwired Server.

Note: Do not begin patching any secondary servers until the primary server has completely restarted all its services after being patched.

a) Extract the patch archive file to a temporary location, for example, `C:\tmp\2.1.3p2`.

b) Log in with administrator privileges, then open a Windows Command Prompt (cmd.exe) and change to the folder where you extracted the patch files.

On Windows 7 and Windows Server 2008R2, start the Command Prompt with the Run as Administrator option.

c) Run `installebf.bat`.

- If the script detects environment issues, you see an error message and the patch process stops. Follow the instructions provided and restart the patch. If you cannot correct the issue, contact Sybase Technical Support.
- If the patch script stops with an unexpected error message, look for locked files in the installation directory tree. Unlock any locked files and re-run the patch script.

Note: The script cannot specifically detect locked files in the installation directory tree. The error message that appears when the script encounters locked files is unpredictable.

d) Press **Y**, then **Enter** to install the patch.

e) When the patch finishes, review the text on your screen for error messages.

On successful completion of the patch, the services for the patched server are started, even if you have specified to manually start services.

Note: After patching the data tier, the first Unwired Server may take as long as an hour to restart.

f) Before proceeding, verify that the patched server is started.

Patching a Microsoft Cluster with Shared Hosts

A Microsoft cluster with shared hosts consists of two Unwired Servers colocated with two data tier servers, with failover capability between data tiers managed by Microsoft Cluster Service, also called Failover Clustering.

1. Shut down the nodes in this order:
   a. Secondary Unwired Servers – shut down these services in this order:
      1. Sybase Unwired Server
      2. Sybase Control Center
   b. Primary Unwired Server – shut down the same services in the same order as for the Secondary Unwired Servers.
2. On the active node, launch the MS Cluster Administrator and make sure the following active Unwired Platform resources are online:
   - Afaria Database Service (available with Unwired Platform version 1.5.2 or earlier)
   - Sybase Unwired CacheDB
   - Sybase Unwired ClusterDB service (available with Unwired Platform version 2.0 or later)
   - Sybase Unwired LogDataDB service (available with Unwired Platform version 2.0 or later)
3. Install the patch on the passive node, with all Unwired Platform resources online on the active node.

   Both data tier and Unwired Server are patched on the passive node.

   Note: Do not begin patching any secondary servers until the primary server has completely restarted all its services after being patched.

   a) Extract the patch archive file to a temporary location, for example, `C:\tmp\2.1.3p2`.
   b) Log in with administrator privileges, then open a Windows Command Prompt (cmd.exe) and change to the folder where you extracted the patch files.

      On Windows 7 and Windows Server 2008R2, start the Command Prompt with the Run as Administrator option.
   c) Run `installebf.bat`.
      - If the script detects environment issues, you see an error message and the patch process stops. Follow the instructions provided and restart the patch. If you cannot correct the issue, contact Sybase Technical Support.
      - If the patch script stops with an unexpected error message, look for locked files in the installation directory tree. Unlock any locked files and re-run the patch script.

Note: The script cannot specifically detect locked files in the installation directory tree. The error message that appears when the script encounters locked files is unpredictable.

d) Press **Y**, then **Enter** to install the patch.

e) When the patch finishes, review the text on your screen for error messages.

On successful completion of the patch, the services for the patched server are started, even if you have specified to manually start services.

Note: After patching the data tier, the first Unwired Server may take as long as an hour to restart.

f) Before proceeding, verify that the patched server is started.

4. Validate the patch on this passive node by starting the Unwired Platform services.

5. For the data tier server, make the passive node active.

a) Launch Microsoft Cluster Administrator.

b) Fail over the active node to the passive node.

The node that was passive is now active.

6. Install the runtime patch on the new passive node.

Both the data tier and Unwired Server are upgraded.

a) Extract the patch archive file to a temporary location, for example, `C:\tmp\2.1.3p2`.

b) Log in with administrator privileges, then open a Windows Command Prompt (cmd.exe) and change to the folder where you extracted the patch files.

On Windows 7 and Windows Server 2008R2, start the Command Prompt with the Run as Administrator option.

c) Run `installebf.bat`.

- If the script detects environment issues, you see an error message and the patch process stops. Follow the instructions provided and restart the patch. If you cannot correct the issue, contact Sybase Technical Support.
- If the patch script stops with an unexpected error message, look for locked files in the installation directory tree. Unlock any locked files and re-run the patch script.

Note: The script cannot specifically detect locked files in the installation directory tree. The error message that appears when the script encounters locked files is unpredictable.

d) Press **Y**, then **Enter** to install the patch.

e) When the patch finishes, review the text on your screen for error messages.

On successful completion of the patch, the services for the patched server are started, even if you have specified to manually start services.

Note: After patching the data tier, the first Unwired Server may take as long as an hour to restart.

f) Before proceeding, verify that the patched server is started.

Patch Readiness Checklist for Sybase Mobile SDK 2.1 ESD #3 Patch 02

Ensure your environment is ready for patching to Sybase Mobile SDK 2.1 ESD #3 Patch 02.

1. Verify that Unwired Platform installations are at version 2.1 ESD #3, or 2.1 ESD #3 Patch 01.
2. Back up the 2.1.3 SDK installation tree.
3. Back up the registry on the SDK installation host.
4. Disable automatic system backups for the 2.1.3 SDK installation tree.
5. Disable virus scan on the installation tree (or disable virus scan altogether).
6. For virtual machine (VM) infrastructure deployments, take a snapshot of each node before proceeding.
7. Adjust User Access Control for your operating system, then validate that the environment continues to work as expected:
   - On Windows 7 or Windows 2008 R2, change the UAC setting for Windows users to **Never Notify** and restart Windows.
   - On Windows Vista and Windows 2008, disable UAC, and restart Windows.
8. Verify that you have administrator privileges on the patch target hosts.
9. Verify that you are using the same Windows user account that was previously used to install the SDK.

Patching to Sybase Mobile SDK 2.1 ESD #3 Patch 02

Use the Sybase Mobile SDK batch file to patch Sybase Mobile SDK 2.1 ESD #3 to version 2.1 ESD #3 Patch 02, retaining full backward compatibility with applications developed in the earlier version of the Mobile SDK.

Prerequisites

Save your important configuration files including the Eclipse workspace and MBO projects.

Task

Note: Sybase Unwired Platform 2.1 ESD #3 Patch 02 is a cumulative patch. You do not need to upgrade to 2.1 ESD #3 Patch 01 prior to installing this patch.

Tip: While the patch script verifies environment readiness, use *Patch Readiness Checklist for Sybase Mobile SDK 2.1 ESD #3 Patch 02* to avoid errors and interruptions in the patch process.

1. Extract the patch archive file to a temporary location, for example, `C:\tmp\2.1.3p2`.
2. Log in with administrator privileges, then open a Windows Command Prompt (cmd.exe) and change to the folder where you extracted the patch files.

   On Windows 7 and Windows Server 2008R2, start the Command Prompt with the Run as Administrator option.
3. Run `installebf.bat`.
   - If the script detects environment issues, you see an error message and the patch process stops. Follow the instructions provided and restart the patch. If you cannot correct the issue, contact Sybase Technical Support.
   - If the patch script stops with an unexpected error message, look for locked files in the installation directory tree. Unlock any locked files and re-run the patch script.

     Note: The script cannot specifically detect locked files in the installation directory tree. The error message that appears when the script encounters locked files is unpredictable.
4. Press **Y**, then **Enter** to install the patch.

Updates to Supported Operating Systems

Sybase Unwired Platform now supports more operating systems.

Unwired Platform 2.1 ESD #3 Patch 02 contains updates which include support for Apple iOS 6.1.x. Hybrid Web Container for iOS released prior to SUP 2.1 ESD #3 Patch 02 does not support Apple iOS 6.0 or later versions.

For a complete list of supported platforms and operating systems, see *Supported Platforms and Development Environments* in the *Supported Hardware and Software* guide.

Updates to Certified Devices

Sybase Unwired Platform now supports more devices.

The iPhone 5, iPad Mini, and iPad with Retina display and A6X chip have been certified for the following applications:

- Object API Applications
- OData SDK Applications
- HTML5/JS Applications—for SUP 2.1 ESD #3 Patch 02 (SUP 2.1 SP03 PL02) only

Fixed Issues in Sybase Mobile SDK 2.1 ESD #3 Patch 02

The 2.1 ESD #3 Patch 02 fixes functionality in Sybase Unwired Platform 2.1 ESD #3.

The following items, identified as Known Issues in the *Release Bulletin 2.1 ESD #3, Patch 01*, are fixed in 2.1 ESD #3 Patch 02. For a complete list of fixes, see the *README* for Sybase Unwired Platform 2.1 ESD #3 Patch 02.

Issue Number	Description
CR 716757	**The `SUPStorage` and shared storage APIs do not work in the iOS 2.1.3 Patch 1 template HWC application.** **Fix**: All instances of `"/sup.amp` were replaced with `"http://localhost/sup.amp` in `SUPStorage.js` in the designer sdk folder.

Known Issues for Unwired Platform Runtime 2.1 ESD #3 Patch 02

Learn about known issues and apply workarounds for Unwired Platform runtime components.

Issue #	Description
RTC-361	**Clear domain logs before 2.1.3 upgrade** **Explanation**: Upgrade to SUP 2.1 ESD#3 will change the DB schema for the domain log DB. If there are too many log entries, the time to complete the upgrade is greatly increased. **Workaround**: Clear all domain log entries before performing the upgrade.
RTC-360	**Backing up the Sybase Unwired Platform Runtime installation fails because the Sybase Control Center path is too long.** **Explanation**: The path to the Embedded Web Container in the `SCC-3_2` folder is too long for Windows to process. **Workaround**: Before backing up the Sybase Unwired Platform Runtime installation, you must delete the contents of *`SUP_HOME`*`\services\EmbeddedWebContainer\container\Jetty-6.1.22\work`. You must delete the contents of this folder from the command prompt, and not from Windows Explorer.

Issue #	Description
RTC-359	**Upgrade cannot proceed because `GWC.war` cannot be deployed.** When running the 2.1.3 upgrade from a command prompt window with a path that has a lowercase drive letter, the update may produce an error that indicates that it was unable to undeploy `GWC.war`. This only affects the listed updates with the specific circumstance where the drive letter is lowercase. **Workaround**: To recover, perform the following steps: **1.** Open a command prompt window with administrator privileges. **2.** Change directories to the *SUP_HOME*`\Servers\UnwiredServer\bin`, where *SUP_HOME* is the install location for Sybase Unwired Platform, typically `C:\Sybase\UnwiredPlatform`. Make sure to specify the drive letter in uppercase. This is required. **3.** Undeploy GWC with the following command: **undeploy.bat webapp-gwc** **4.** If there are no errors, delete *SUP_HOME*`\Servers\UnwiredServer\deploy\webapps\gwc`, if it exists, with the following command: **rmdir /S /Q** *SUP_HOME*`\Servers\UnwiredServer\deploy\webapps\gwc` **5.** Deploy the new GWC with the following command: **deploy.bat** *SUP_HOME*`\Servers\UnwiredServer\temp\GWC.war` **6.** Undeploy notifications with the following command: **undeploy.bat webapp-notifications** **7.** If there are no errors, delete *SUP_HOME*`\Servers\UnwiredServer\deploy\webapps\notifications`, if it exists, with the following command: **rmdir /S /Q** *SUP_HOME*`\Servers\UnwiredServer\deploy\webapps\notifications` **8.** Deploy the new notifications WAR file with the following command: **deploy.bat** *SUP_HOME*`\Servers\UnwiredServer\temp\notifications.war`

Issue #	Description

CR 729104	**DCN JSON parsing error**
	
	Explanation: When trying to push an upsert to the CDB, where a string has a special value (such as % or &) a parsing error is thrown. According to World Wide Web Consortium (W3C) and Internet Engineering Task Force (IETF) specifications, url and form data must be encoded before being sent to the server. Only alphanumerics [0-9, a-z, A-Z], the special characters contained in the parentheses ($ - . ! * ' () ,), and reserved characters used for specific reserved purposes may be used unencoded within a URL. Please refer to *http://www.w3.org/TR/1999/REC-html401-19991224/interact/forms.html#h-17.13.4.1* and *http://www.ietf.org/rfc/rfc2045.txt* for details.
	
	Workaround: The Unwired Server takes uft-8 as a URL encoded charset. So, if the client sends a request using either the `GET` or `POST` method, the request content must be encoded by utf-8 according to the URL encoding rules. For example, the URL `http://serverhost:8000/dcn/DCNServlet?cmd=dcn&username=supAdmin&password=s3pAdmin&domain=default&package=sup101:1.0&dcn_request={"pkg":"dummy","messages":[{"id":"4","mbo":"Customer","op":":upsert","cols":{"id":"10004","fname":"B%B"}}]}` has reserved and special characters {} [] : " %. These chars should be encoded.
	
	If you use java to make the DCN request, use `URLEncoder.encode(parameter,"UTF-8")` to encode your URL query parameter or post form data before you make the request. For example, if you construct the URL listed in the previous example, you would use the following code:
	
	`String url="http://serverhost:8000/dcn/DCNServlet?cmd=dcn&username=supAdmin&password=s3pAdmin&domain=default";`
	`url+="&package="+URLEncoder.encode("sup101:1.0","UTF-8");`
	`url+="&dcn_request="+URLEncoder.encode("sup101:1.0","{\"pkg\":\"dummy\",\"messages\":[{\"id\":\"4\",\"mbo\":\"Customer\",\"op\":\":upsert\",\"cols\":{\"id\":\"10004\",\"fname\":\"B%B\"}}]}");`

Known Issues for Sybase Mobile SDK 2.1 ESD #3 Patch 02

Learn about known issues and apply workarounds for Sybase Mobile SDK 2.1 ESD #3 Patch 02.

Issue Number	Description
RTC-367	**The internal default page size is used during Object API code generation if Unwired Server is not selected, which can lead to code generation failure without visible error from the code generation wizard.** If the MBO developer does not select Unwired Server during code generation, the default page size is used. If the internal default page size is too small, code generation fails without creating objectAPI code, and the page size error only displays in the Error Log view, but not in the code generation wizard. **Workaround**: Select an Unwired Server connection profile to enable the **Page size** field, then select a proper page size. Also, as a common troubleshooting practice, open the Error Log view in Unwired WorkSpace to find clues when the MBO developer detects any error symptoms.
RTC-338	**MBO project fails to deploy after renaming MBO with custom object queries** After an MBO with custom object queries is renamed, the MBO project deployment fails with an Unknown Error exception. **Workaround:** Modify the object query definition to reference the new MBO name.
CR 722779	**Hybrid Workflow Application does not use the full screen on iPhone5** The Hybrid Workflow Application does not use the full screen on iPhone5. In portrait mode, narrow black bars appear on the top and bottom of the screen. In landscape mode, narrow black bars appear along the left and right side of the screen. **Workaround:** None.

Issue Number	Description

CR 721724	**The MAKit 2.1.3 Code Example orientation freezes in Apple iOS 6.0.**
	When attempting to change the orientation from landscape to portrait, or from portrait to landscape on the iPad2, the MAKit Code Example does not change orientation.
	Workaround: To run the MAKit 2.1.3 in iOS6.0 without any orientation issues:
	1. In `MAKitCodeExamplesAppDelegate` replace the line
	`[self.window addSubview:navController.view];`
	with
	`[set.window setRootViewController:navController];`
	2. In the `MAKitCodeExamples-info.plist` file, delete the sub items (item 0 and item 1) under the "Supported interface orientations".

CR 717094	**Exception thrown on BlackBerry 9860 with BlackBerry OS 7.0** When attempting to remove a persistent object on a BlackBerry 9860 device with BlackBerry OS 7.0, a **ControlledAccessException** occurs. **Workaround**: None. The failure to delete the object is not a catastrophic failure. The event is logged and otherwise ignored.

Documentation Issues and Updates for Unwired Platform Runtime 2.1 ESD #3 Patch 02

Read about updates, corrections, and clarifications to the documentation released with Sybase Unwired Platform Runtime 2.1 ESD #3 Patch 02.

Issue Number	Description
N/A	**Security configuration does not require LDAP provider to authenticate logins** In the first paragraph of the *NTProxy Security Provider* topic in *Security*, this second sentence is incorrect and should be disregarded: Add an LDAP provider to a security configuration to authenticate administrator logins (on the "admin" security configuration on the "default" domain) or device user logins (any custom security configuration for that purpose). *NTProxy Security Provider* is found in: • *Security > Server Security > Enabling Authentication and RBAC for User Logins > Authentication in Unwired Platform > Built-in Security Providers for User Authentication and Authorization*

Issue Number	Description
N/A	**Domain names are case insensitive** The Creating and Enabling a New Domain topic in *System Administration* should state that domain names are case insensitive. To read the note in context, see *Creating and Enabling a New Domain* on page 16.
RTC-366	**Missing info in Update Properties (updateprops.bat) Utility topic** Two notes were added to *Update Properties (updateprops.bat) Utility*, located in *System Administration*: • Unless documented otherwise, use Sybase Control Center to change most properties in the runtime to avoid unnecessary complication. • The -r, -x, and -f options are reserved for internal product use only. While these options are supported by the command line, they are not intended for administrator use. You can read an updated version of the topic in this release bulletin. See *Update Properties (updateprops.bat) Utility* on page 17.
CR 736637	**Cannot add custom JCO properties** You cannot add custom JCO connection properties. See *http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.infocenter.dc01332.0213/doc/html/dst1255703078075.html* for more information about SAP Java Connector Properties.
CR 717585	**Binary Personalisation parameter has limits on data size** In *Sybase Unwired Workspace - Mobile Business Object Development* > *Develop* > *Working with Mobile Business Objects* > *Mobile Business Object Data Properties* > *Managing Persionalization Keys* > *Personalization Key Guidelines and Limitations* there is additional information: When assigning a personalization key that uses a binary datatype to an operation argument or load argument with BIGBINARY type, if the data in the personalization key exceeds 35K, the server throws exceptions during synchronization: `[LNSupController+Finalize.m:61] Send data ExceptionSUPPersistenceException: SUPPersistenceException from synchronize: -- SUPSynchronizeException: loginFail,Sync failed: -1497 (ERROR) %1:4 %2:4000` When a binary personalization is assigned to an operation argument or a load argument, its size cannot exceed 35K. Additionally it is not recommended to map a personalization key to big object types such as `BIGSTRING` or `BIGBINARY`.

Creating and Enabling a New Domain

Create and configure multiple domains within a single Unwired Platform installation. A domain must be enabled for application users to access the packages deployed in the domain.

Enabling a domain also triggers synchronization of the domain changes to the secondary nodes in the cluster. Application users who attempt to access a disabled domain receive an error message.

Prerequisites

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

Task

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

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

Update Properties (updateprops.bat) Utility

Updates cluster or server properties.

Note: Unless documented otherwise, use Sybase Control Center to change most properties in the runtime to avoid unnecessary complication.

Syntax

```
updateprops.bat [-u username] [-p password] [-d dsn]
[-cn clusterName] [-nv "<propertyName=NewValue>"] [-v]
```

Parameters

- **-u *username* –** the platform administrator username.
- **-p *password* –** the platform administrator password.
- **-d *dsn* –** the data source name (DSN) of the cluster database.
- **-cn *clusterName* –** the name that identifies the Unwired Platform Cluster

- **-nv "<*propertyName=NewValue*>" –** one or more platform property values that requires change. Multiple values can be defined; however, they must be separated by the pound symbol (#). For example:

  ```
  –nv
  "ml.threadcount=10#sup.admin.port=2005#sup.sync.port=2490"
  ```
- **-v –** use verbose output in the command window.

Note: The -r, -x, and -f options are reserved for internal product use only. While these options are supported by the command line, they are not intended for administrator use.

Examples

- **Changing a cdb threadcount property –** Update the ml.threadcount property of the production environment cache database to `20` by running:

  updateProps.bat -nv "ml.threadcount=20"

  This is only recommended for deployment editions of Unwired Platform.

Usage

Before running this utility, ensure that the data tier is available; otherwise platform data is not modified correctly.

Documentation Issues and Updates for Sybase Mobile SDK 2.1 ESD #3 Patch 02

Read about updates, corrections, and clarifications to the documentation released with Sybase Mobile SDK 2.1 ESD #3 Patch 02.

Issue Number	Description
NA3-2386	**Delete container files when closing hybrid app** The last sentence of the *Hybrid Web Container Files* section in *Developer Guide: Mobile Workflow Packages* > *Mobile Workflow Development* > *Security* > *Content Security on Devices*, is changed to read: "The temp files are removed when the mobile workflow application is closed."

Issue Number	Description
NA2-1743	**Corrected user authentication information** In the topic *Logging In*, the second paragraph incorrectly lists the startConnection method as an appropriate method for user authentication. The paragraph should read: Authenticate the user for data synchronization by calling the generated database API onlineLogin method. This topic is found in: • *Developer Guide: Android Object API Applications > Developing the Application Using the Object API > Initializing an Application > Initially Starting an Application* • *Developer Guide: BlackBerry Object API Applications > Developing the Application Using the Object API > Initializing an Application > Initially Starting an Application* • *Developer Guide: iOS Object API Applications > Developing the Application Using the Object API > Initializing an Application > Initially Starting an Application* • *Developer Guide: Windows and Windows Mobile Object API Applications > Developing the Application Using the Object API > Initializing an Application > Initially Starting an Application*
CR 730839	**REST Service endpoint does not support JSON** In the topic *REST Web Services* in *Sybase Unwired WorkSpace - Mobile Business Object Development* the statement: "Supports resources with multiple representations (Transfer XML, JavaScript Object notation(JSON), or both)" is not correct. The REST Service endpoint does not support JSON, so resources with multiple representations are not supported.

Issue Number	Description
CR 730110	**When executing synchronization on a group, all MBOs in submitPending state will be synchronized.** In the topic *Performing Mobile Business Object Synchronization*, there is additional information: In the upload phase of synchronizaion, all of the pending MBOs will be uploaded from client to server, even if the synchronization is intended only for a specific synchronization group. This topic is found in: • *Developer Guide: Android Object API Applications > Client Object API Usage > Synchronization APIs* • *Developer Guide: BlackBerry Object API Applications > Client Object API Usage > Synchronization APIs* • *Developer Guide: BlackBerry Object API Applications (with DOE) > Client Object API Usage > Synchronization APIs* • *Developer Guide: iOS Object API Applications > Client Object API Usage > Synchronization APIs* • *Developer Guide: Windows and Windows Mobile Object API Applications > Client Object API Usage > Synchronization APIs*

Release Bulletin 2.1 ESD #3 Patch 01 (SUP 2.1 SP03 PL01)

The 2.1 ESD #3 Patch 01 (SUP2.1 SP03 PL01) provides improved functionality and fixes for Sybase® Unwired Platform 2.1 ESD #3.

Note: The newest patch, Sybase Unwired Platform 2.1 ESD #3 Patch 02 provides a fix for a known issue (CR 716757) in this patch (2.1 ESD #3 Patch 01).

Functionality improvements:

- Customizing the Hybrid Web Container for Android and iOS is now simpler and easier.
- You can set a URL suffix for the server connection URL for Android, BlackBerry, and iOS OData applications.
- OData clients can receive HTTP error codes through a listener.

For fixes to known issues, see *Fixed Issues in Patch 01* on page 26. For patch contents, and a complete list of fixes, see the *README* for Sybase Unwired Platform 2.1 ESD #3 Patch 01.

Improved Functionality for Hybrid Web Container in 2.1 ESD #3 Patch 01

Sybase Unwired Platform 2.1 ESD #3 Patch 01 offers improvements to the Hybrid Web Container.

Customizing the Hybrid Web Container for Android and iOS is now simpler and easier. See *Developer Guide: Mobile Workflow Packages > Mobile Workflow Development > Hybrid Web Container Customization*.

Improved Functionality for OData SDK in 2.1 ESD #3 Patch 01

Sybase Unwired Platform 2.1 ESD #3 Patch 01 provides several improvements to the OData SDK.

- You can set a URL suffix for the relay server connection URL for Android, BlackBerry, and iOS OData applications.
- There are 2 new HTTP error codes that the server can return to a client during a request-response cycle.
- OData clients can receive HTTP error codes from a network edge by implementing a listener.

Setting the URL Suffix for Android Applications

You can set the URL suffix for the Relay Server connection URL.

Syntax

This API belongs to the `LiteUserManager` class.

```
public void setRelayServerURLTemplate(String URLSuffix)
```

Parameters

- **URLSuffix –** Corresponds to the URL suffix configured for the Relay Server.

Examples

- **Set the URL suffix**

```
LiteUserManager lurm = LiteUserManager.getInstance();
lurm.setRelayServerUrlTemplate("http/suffix");
```

Setting the URL Suffix for BlackBerry Applications

You can set the URL suffix for the relay server connection URL .

Syntax

This API belongs to the `UserManager` class.

```
public static void setRelayServerURLTemplate(String URLSuffix)
```

Parameters

- **URLSuffix –** Corresponds to the URL suffix configured for the Relay Server.

Examples

- **Set the URL suffix**

```
UserManager.setRelayServerURLTemplate("sk/sr");
```

Setting the URL Suffix for iOS Applications

You can set the URL suffix for the relay server connection URL.

Syntax

This API belongs to the `LiteSUPUserManager` class.

```
- (void) setRelayServerUrlTemplate: (NSString *) _urlSuffix
```

Parameters

- **_urlSuffix –** corresponds to the URL suffix configured for the Relay Server.

Examples

- **Set the URL suffix**

```
LiteSUPUserManager* manager = [LiteSUPUserManager
getInstance:@"NewFlight"];
[manager setRelayServerUrlTemplate:@"/ee/hrg/wgwrgw.dll"];
```

HTTP Error Codes for ODP Client Requests

Unwired Server returns new HTTP error codes to the client based on some scenarios defined for online applications.

Table 1. HTTP Error Codes

Error Code	Probable Cause	Manual Recovery Action
403	If a client application requests an EIS URL that is not registered in Sybase Control Center, the server rejects the request.	Register the EIS URL in Sybase Control Center.
405	When the back end updates data on the device, the server supports only an HTTP POST request method. For any other request method used by the back end, the server throws an error.	N/A

Registering a Listener for HTTP Error in Android Applications

To ensure that OData Android clients are notified of HTTP errors while establishing a connection with the network edge, implement a listener.

Syntax

```
public static void setODPHTTPErrorListener(IODPHttpErrorListener
oListener) throws MessagingClientException
```

Parameters

- **oListener –** listener object that implements the interface `IODPHttpErrorListener`.

Examples

- **Implement the listener**

```
public class UserRegistration{

     public void startUserRegistration() {
       UserManager.initialize(appID);
       UserManager.setConnectionProfile(serverIP, serverPort,
farmID);
       UserManager.enableHTTPS(true);
      ODPErrorListener odpErrorListener = new ODPErrorListener();

LiteMessagingClient.setODPHTTPErrorListener(odpErrorListener);
       UserManager.registerUser(username, securityConfig,
```

```
password);
     }

public class ODPErrorListener implements IODPHttpErrorListener
{
    // callback method for HTTP Error Code Listener
    @Override
   public void onHttpError(int iErrorCode, String sErrorMessage,
Hashtable oHeaders)
     {
        // TODO Auto-generated method stub
        Log.i("MDP", "Error info" +iErrorCode+sErrorMessage);
      }
}
```

Registering a Listener for HTTP Error in BlackBerry Applications

To ensure that OData BlackBerry clients are notified of HTTP errors while establishing a connection with the network edge, implement a listener.

Syntax

```
public static void setODPHTTPErrorListener(IODPHttpErrorListener
oListener) throws MessagingClientException
```

Parameters

- **oListener –** listener object that implements the interface `IODPHttpErrorListener`.

Examples

- **Implement the listener**

```
public class UserRegistration implements
ODPClientListeners.IODPHttpErrorListener{
 .
 .
   public void startUserRegistration(){
     UserManager.initialize(appID);
     UserManager.setConnectionProfile(serverIP,serverPort,
farmID);
     UserManager.setODPHttpErrorListener(this);
     UserManager.registerUser(username,securityConfig,password);
     }

  //callback method for HttpError
  public void onHttpError(int errorCode, String errorMsg,
Hashtable errorHeader) {
    logger.info(null, "On HttpError", "Error Info" +errorCode
+errorMsg);
  }
```

```
}
```

Registering a Listener for HTTP Errors in iOS Applications

To ensure that OData iOS clients are notified of HTTP errors while establishing a connection with the network edge, implement a listener.

Syntax

To register a listener for HTTP error codes, implement the protocol `ODPHTTPErrorListenerDelegate`.

```
@protocol ODPHTTPErrorListenerDelegate <NSObject>
@required
- (void) onHTTPError: (int)code errorMessage:(NSString*)message
httpHeaders:(NSDictionary*)headers;
```

Examples

- **Register a delegate for HTTP authentication challenge**

```
-(void) onHTTPError:(int)code errorMessage:(NSSTring*)message
httpHeaders:(NSDictionary*)headers
{
  if (code==xxx){
   //Display Error Messages
  }
}

-(IBAction)registerUser:(id)sender
{
 LiteSUPUserManager* userManager = nil;
 @try
 {
  userManager = [LiteSUPUserManager getInstance:@"NewFlight"];
  [ODPClientListeners setHTTPErrorListenerDelegate:self];
  [userManager setConnectionProfile:@<host> withSupPort:<port>
withServerFarmID:@<farmID>];
  [userManager registerUser:@<supuser>
withSecurityConfig:@<securityconfig> withPassword:@<password>];
 }
 @catch (NSException *exception {
  NSLog(@"%@:%@", [[exception userInfo]
objectForKey:@"ErrorCode"],[[exception userInfo]
objectForKey:@"ErrorMessage"]);
 }
}
```

Fixed Issues in 2.1 ESD #3 Patch 01

The 2.1 ESD #3 Patch 01 fixes functionality in Sybase Unwired Platform 2.1 ESD #3.

The following items, identified as Known Issues in the *Release Bulletin 2.1 ESD #3*, are fixed in 2.1 ESD #3 Patch 01. For a complete list of fixes, see the *README* for Sybase Unwired Platform 2.1 ESD #3 Patch 01.

Issue Number	Description
CR 709396	**For Online Data Proxy, any back-end datasource could push data to Unwired Server without authentication.** **Fix**: For any back-end datasource to push data to Unwired Server, it must use SUP DCN user credentials for authentication. **References**: See *SUP DCN User Role* in *Sybase Control Center for Sybase Unwired Platform* and *Mapping Roles for Domains, Packages, or Applications* in *Security*.
CR 709392	**Missing error code methods for the BlackBerry client in com.sybase.persistence.SynchronizationContext.** This was a known issue for native Object API. **Fix**: The BlackBerry client now has the following Object API methods for getting and settings error codes and messages in `com.sybase.persistence.SynchronizationContext`: • `getErrorCode` • `setErrorCode` • `getErrorMessage` • `setErrorMessage`
CR 709335	**For Online Data Proxy, any back-end datasource could communicate with Unwired Server without being registered in Sybase Control Center.** **Fix**: In addition to the application endpoint, you must register any URL that is required by an application for a proxy service to enable communication with Unwired Server. **References**: See *Proxy Properties* on page 27.

Issue Number	Description
CR 707616	**When writing a Hybrid Web Application for iOS using the PhoneGap API `navigator.camera.getPicture()` and specifying `FILE_URI` as the destination type, you cannot display the image**. The same is true for the result of any call to `FileEntry.toURI()`, `DirectoryEntry.toURI()`, or anything else that references file URIs. This was a known issue for Hybrid Web Container. **Fix**: The library now loads files using a file URI. This allows the workflow to access other URIs and make http requests to the native code. **References**: See *Content Security on iOS Devices* in *Developer Guide: Mobile Workflow Packages*.
CR 704758	**Cannot delete replication devices in Sybase Control Center; license information incorrect.** This was a known issue for Sybase Control Center. **Fix**: • If the RBS application client runtime version is older than 2.1.2, delete an MBO package user to free up the license as long as the replication device has no other package users registered for other MBO packages. • If the RBS application client runtime version is 2.1.2 or later, delete the application connection.

Proxy Properties

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

Name	Description	Supported Values
User	Not currently used.	
Certificate Alias	Not currently used.	
Address	Corresponds to the application endpoint provided when registering an application.	Must be a valid application endpoint.
Pool Size	Determines the maximum number of connections allocated to the pool for this datasource.	The default value set for the pool size is 25.
Password	Not currently used.	

Note:

- In Sybase Control Center, when the application endpoint for a registered application is modified under the **Applications** node, you must manually update the **Address** in the proxy properties of the connection pool.
- In Sybase Control Center, in addition to the application endpoint, you must register any URL that is required by an application for a proxy service to enable communication with Unwired Server.

Known Issues for Unwired Platform Runtime 2.1 ESD #3 Patch 01

Learn about known issues and apply workarounds for Unwired Platform Runtime 2.1 ESD #3 Patch 01.

Issue Number	Description
SMPONP-13039	**Oracle support for JDK ended December 31, 2013** The support agreement between Oracle and SAP is no longer in effect as of December 31, 2013. **Workaround**: See *SAP Note 1949332* for workaround and additional instructions.
CR 714677	**Client application subscribe request fails.** If a client application uses the manual registration method, and its application connection is registered using the Unwired Server Management API `SUPApplication.registerApplicationConnections()`, the client application subscribe request fails. The error message is: `Application connection {0} is not for appId {1}.` **Workaround**: None. Contact SAP® Support for further help.
CR 714679	**SCC freezes up with "Connecting with the SCC Managed Object Server" message after login** **Workaround**: Instead of using the standard URL (`https:// < hostname > :8283/scc`), if accessing from the same host use `https://localhost:8283/scc` or `https:// < host name.domain > :8283/scc`. Another option is to try accessing from a remote machine.

Documentation Issues and Updates for Unwired Platform Runtime 2.1 ESD #3 Patch 01

Read about updates, corrections, and clarifications to the documentation released with Sybase Unwired Platform Runtime 2.1 ESD #3 Patch 01.

Issue Number	Description
CR 712469	This topic is not currently included in the *Troubleshooting Guide*. *Application Doesn't Match the Package* on page 29

Application Doesn't Match the Package

A typical scenario is when you modify one of the mobile application projects in a multipackage device application (a multimoblie application project), and redeploy using "Replace" mode. Sometimes one mobile application project is missing, resulting in the error message `Application doesn't match the package`.

Explanation: This message indicates the device application is accessing an Mobile Business Object (MBO) package that is currently not assigned to its application ID.

Solution: Make sure the MBO packages accessed by the application are assigned to the application ID. You can fix the problem using Sybase Control Center, or Unwired WorkSpace:

1. Add the missing application using either application:
   - From Sybase Control Center, navigate to the Applications node, and use the Properties button to add the MBO package.
   - From Unwired WorkSpace, open the Properties dialog in the MBO Package Eclipse project, access the Mobile Application Project page, and add the `applicationid` to the list of applications.
2. Redeploy the application.

Known Issues for Sybase Mobile SDK 2.1 ESD #3 Patch 01

Learn about known issues and apply workarounds for Sybase Mobile SDK 2.1 ESD #3 Patch 01.

Issue Number	Description
SMPONP-6352	**BB 9800 throws exception message "The memory available on your device is low. Close some of the items below."** Using the BlackBerry HWC on older devices may result in warnings/errors from the operating system, such as "The memory available on your device is low. Close some of the items below." **Workaround**: Run the BlackBerry HWC on newer OS 6.x or OS 7.x devices with more memory. The underlying problem is a memory leak in the native WebKit engine in the RIM OS. A fix for this is not available from RIM for Java based platforms. *https://www.blackberry.com/jira/browse/JAVAAPI-2275*
CR 716757	**The `SUPStorage` and shared storage APIs do not work in the iOS 2.1.3 Patch 1 template HWC application.** **Workaround**: Find `SUPStorage.js` in the designer sdk folder and replace all instances of **"**`/sup.amp` with **"**`http://localhost/sup.amp`.
CR 715053	**Issue**: **iOS SUPQuery LEFT_OUTER_JOIN works as INNER_JOIN.** **Workaround**: None.
CR 713043	**Issue**: **Foreign key constraint violation error message.** A foreign key constraint violation error message occurs if you create an MBO by dragging and dropping a table in which the foreign key and the primary key are both 'identity' types. **Workaround**: The MBO is created, and you can safely ignore these error messages.

Documentation Issues and Updates for Sybase Mobile SDK 2.1 ESD #3 Patch 01

Read about updates, corrections, and clarifications to the documentation released with Sybase Mobile SDK 2.1 ESD #3 Patch 01.

Issue Number	Description
CR 713636	**Installation path has changed for Object API iOS files.** In the *Tutorial: iOS Object API Application Development*, the installation paths are not updated to reflect the new installation path structure for the patch. The correct installation path is `\Unwired Platform\MobileSDK213\ObjectAPI \iOS`.
CR 709826	**Clarification for onReplaySuccess:(id)entityObject.** In the *Developer Guide: iOS Object API Applications* > *Client Object API Usage* > *Callback and Listener APIs*, there is additional information: **Note:** The `onReplaySuccess:(id)entityObject` is an MBO object instance that contains the data prior to the synchronization. You can use the Change Log API to find records that occur after the synchronization.
CR 708707	**Best practice for deleting the client database.** In *Client Object API Usage* > *Synchronization Profile* > *Asynchronous Operation Replay* in the developer guides, there is additional information: By default, asynchronous operation replay is enabled. When the application is connected (by `Application.StartConnection()` or `Application.RegisterApplication`), it may receive background notifications and trigger a `synchronize` or other database operation. If you try to delete the database, you may receive database exceptions. Before deleting the database, stop the application connection (`Application.StopConnection()`).

Release Bulletin 2.1 ESD #3

The Unwired Platform 2.1 ESD #3 upgrades Unwired Platform 2.1 with new features and functionality.

Upgrading to Unwired Platform Runtime 2.1 ESD #3

You can upgrade to Unwired Platform 2.1 ESD #3 from Unwired Platform 2.1. If your current version is earlier than 2.1, you must upgrade to 2.1 before proceeding.

The upgrade installer automatically adds Sybase® SAP® DOE Connector (DOE-C) if it is not present in your earlier version installation, and upgrades DOE-C if it is present.

Upgrade Readiness Checklist

Ensure your environment is ready for upgrade.

- Run a full backup of each node in your system infrastructure.
- Back up the registry on each node in your system infrastructure.
- Back up the shared database, if you have Unwired Platform installed in a cluster.
- Make sure that the available disk space on each data tier node in a cluster, or on a single-server installation, is sufficient to accommodate a 50% increase in the size of the shared database.
- For virtual machine (VM) infrastructure deployments, take a snapshot of each node before proceeding.
- Adjust User Access Control for your operating system, then validate that the environment continues to work as expected:
  - On Windows 7 or Windows 2008 R2, change the UAC setting for Windows users to Never Notify, and restart Windows.
  - On Windows Vista and Windows 2008, disable UAC, and restart Windows.
- Verify that Unwired Platform installations are at version 2.1 or later.
- Verify that you have administrator privileges on the upgrade target hosts.
- Verify that you are using the same Windows user account that was previously used to install Unwired Platform.
- Verify that database administrator user names and passwords for the configurable Unwired Platform databases have not changed from the default settings ("dba"/"sql"), or enter the new values here:

Database	Admin User Name	Password
Sybase Unwired CacheDB		
Sybase Unwired ClusterDB		
Sybase Unwired LogDataDB		
Sybase Unwired MessagingDB		

Upgrading a Single Node

In a single-node installation, all Unwired Platform server components are located on a single server.

The upgrade script verifies environment readiness, but you can use the readiness checklist to avoid errors and interruptions in the upgrade process.

1. Extract the patch archive file to a temporary location, for example, `C:\tmp\2.1.3`.
2. Logging in with administrator privileges, open a Windows Command Prompt (cmd.exe) and change to the folder where you extracted the upgrade files.
3. Run `installebf.bat`.

   The script runs an initial validation check on the files extracted from the upgrade package. If files are corrupted, delete the corrupted archive and download another copy of the upgrade package.

   If the script detects environment issues, you see an error message and the upgrade process stops. Follow the instructions provided and restart the upgrade. If you cannot correct the issue, contact Sybase Technical Support.

   The script does not detect locked files in the installation directory tree. The error message that appears when the script encounters locked files varies.

   If the installer stops with an unexpected error message, look for locked files in the installation directory tree. Unlock any locked files you find and re-run the upgrade script.
4. Press **Y**, then **Enter** to install the upgrade.

Upgrading a Simple Cluster

A simple cluster consists of one data tier node and one or more Unwired Server nodes, with failover capability between multiple Unwired Server nodes.

1. Shut down the nodes in this order:
   a. Secondary Unwired Servers – shut down services in this order:
      1. Sybase Unwired Server
      2. Sybase Control Center
      3. Sybase Unwired SampleDB (not applicable to Enterprise Server Edition)
   b. Primary Unwired Server – shut down services in this order:
      1. Sybase Unwired Server
      2. Sybase Control Center
      3. Sybase Unwired SampleDB (not applicable to Enterprise Server Edition)
   c. Data tier server – shut down these services, in any order:
      - Sybase Unwired CacheDB
      - Sybase Unwired ClusterDB

- Sybase Unwired LogDataDB
- Advantage Database Server®

2. Install the upgrade on each node in the cluster, in this sequence:
   a. Data tier node
   b. Primary Unwired Server
   c. Secondary Unwired Servers

Note: Do not begin upgrading any secondary servers until the primary server has completely restarted all its services after being upgraded.

a) Extract the patch archive file to a temporary location.
b) Logging in with administrator privileges, open a Windows Command Prompt (cmd.exe) and change to the folder where you extracted the upgrade files.
c) Run `installebf.bat`.

   The script runs an initial validation check on the files extracted from the upgrade package. If files are corrupted, delete the corrupted archive and download another copy of the upgrade package.

   If the script detects environment issues, you see an error message and the upgrade process stops. Follow the instructions provided and restart the upgrade. If you cannot correct the issue, contact Sybase Technical Support.

   The script does not detect locked files in the installation directory tree. The error message that appears when the script encounters locked files varies.

   If the installer stops with an unexpected error message, look for locked files in the installation directory tree. Unlock any locked files you find and re-run the upgrade script.
d) Press **Y**, then **Enter** to install the upgrade.
e) When the upgrade finishes, review the text on your screen for error messages.

   On successful completion of the upgrade, the services for the upgraded server are started, even if you have specified that services for the server are to be manually started.

   Note: After upgrading the data tier, the first Unwired Server may take as long as an hour to restart.
f) Before proceeding, verify that the upgraded server is started.

Upgrading a Microsoft Cluster

A Microsoft cluster consists of two data tier nodes and multiple Unwired Server nodes, with failover between data tier nodes managed by Microsoft Cluster Service, also called Failover Clustering.

1. Shut down the nodes in this order:
   a. Secondary Unwired Servers – shut down these services in this order:

1. Sybase Unwired Server
2. Sybase Control Center
3. Sybase Unwired SampleDB (not applicable to Enterprise Edition)

b. Primary Unwired Servers – shut down these services in this order:

1. Sybase Unwired Server
2. Sybase Control Center
3. Sybase Unwired SampleDB (not applicable to Enterprise Edition)

2. On the active data tier node, launch the MS Cluster Administrator and make sure the following active Unwired Platform resources are offline:

- Advantage Database Server service
- Afaria Database Service (available with Unwired Platform version 1.5.2 or earlier)
- Sybase Unwired CacheDB service
- Sybase Unwired ClusterDB service (available with Unwired Platform version 2.0 or later new installations)
- Sybase Unwired LogDataDB service (available with Unwired Platform version 2.0 or later new installations)

3. Install the upgrade on the active data tier node:

a) Extract the patch archive file to a temporary location.

b) Logging in with administrator privileges, open a Windows Command Prompt (cmd.exe) and change to the folder where you extracted the upgrade files.

c) Run `installebf.bat`.

The script runs an initial validation check on the files extracted from the upgrade package. If files are corrupted, delete the corrupted archive and download another copy of the upgrade package.

If the script detects environment issues, you see an error message and the upgrade process stops. Follow the instructions provided and restart the upgrade. If you cannot correct the issue, contact Sybase Technical Support.

The script does not detect locked files in the installation directory tree. The error message that appears when the script encounters locked files varies.

If the installer stops with an unexpected error message, look for locked files in the installation directory tree. Unlock any locked files you find and re-run the upgrade script.

d) Press **Y**, then **Enter** to install the upgrade.

e) When the upgrade finishes, review the text on your screen for error messages.

On successful completion of the upgrade, the services for the upgraded server are started, even if you have specified that services for the server are to be manually started.

Note: After upgrading the data tier, the first Unwired Server may take as long as an hour to restart.

f) Before proceeding, verify that the upgraded server is started.

4. Validate the upgrade on this node by bringing the stopped Microsoft cluster resources online and ensuring they are restarted.
5. Make the passive node active and ensure all database services are offline.
   a) Launch Microsoft Cluster Administrator.
   b) Fail over the active node to the passive node.

      The node that was passive is now active.
6. Install the upgrade on the new active data tier node:
   a) Extract the patch archive file to a temporary location.
   b) Logging in with administrator privileges, open a Windows Command Prompt (cmd.exe) and change to the folder where you extracted the upgrade files.
   c) Run `installebf.bat`.

      The script runs an initial validation check on the files extracted from the upgrade package. If files are corrupted, delete the corrupted archive and download another copy of the upgrade package.

      If the script detects environment issues, you see an error message and the upgrade process stops. Follow the instructions provided and restart the upgrade. If you cannot correct the issue, contact Sybase Technical Support.

      The script does not detect locked files in the installation directory tree. The error message that appears when the script encounters locked files varies.

      If the installer stops with an unexpected error message, look for locked files in the installation directory tree. Unlock any locked files you find and re-run the upgrade script.
   d) Press **Y**, then **Enter** to install the upgrade.
   e) When the upgrade finishes, review the text on your screen for error messages.

      On successful completion of the upgrade, the services for the upgraded server are started, even if you have specified that services for the server are to be manually started.

      Note: After upgrading the data tier, the first Unwired Server may take as long as an hour to restart.
   f) Before proceeding, verify that the upgraded server is started.
7. Install the upgrade on the Unwired Server nodes, in this sequence:
   - Primary Unwired Server – shut down services in this order:
     a. Sybase Unwired Server
     b. Sybase Control Center
     c. Sybase Unwired SampleDB (not applicable to Enterprise Server Edition)
   - Secondary Unwired Servers – shut down same services in same order as for the primary Unwired Server.

Note: Do not begin upgrading any secondary servers until the primary server has completely restarted all its services after being upgraded.

a) Extract the patch archive file to a temporary location.
b) Logging in with Administrator privileges, open a Windows Command Prompt (cmd.exe) and change to the folder where you extracted the upgrade files.
c) Run `installebf.bat`.

   The script runs an initial validation check on the files extracted from the upgrade package. If files are corrupted, delete the corrupted archive and download another package.

   If the script detects environment issues, you see an error message and the upgrade process stops. Follow the instructions provided and restart the upgrade. If you cannot correct the issue, contact Sybase Technical Support.

   The script does not detect locked files in the installation directory tree. The error message that appears when the script encounters locked files is unpredictable.

   If the installer stops with an unexpected error message, look for locked files in the installation directory tree. Unlock any locked files you find and re-run the upgrade script.
d) Press **Y**, then **Enter** to install the upgrade.
e) Enter the Windows service account name and password.
f) When the upgrade finishes, review the text on your screen for error messages.

   On successful completion of the upgrade, the services for the upgraded server are started, even if you have specified that services for the server are to be manually started.

   Note: After upgrading the data tier, the first Unwired Server may take as long as an hour to restart.
g) Before proceeding, verify that the upgraded server is started.

Upgrading a Microsoft Cluster with Shared Hosts

A Microsoft cluster with shared hosts consists of two Unwired Servers colocated with two data tier servers, with failover capability between data tiers managed by Microsoft Cluster Service, also called Failover Clustering.

1. Shut down the nodes in this order:
   a. Secondary Unwired Servers – shut down these services in this order:
      1. Sybase Unwired Server
      2. Sybase Control Center
   b. Primary Unwired Server – shut down these services in this order:
      1. Sybase Unwired Server
      2. Sybase Control Center

2. On the active node, launch the MS Cluster Administrator and make sure the following active Unwired Platform resources are online:
   - Advantage Database Server service
   - Afaria Database Service (available with Unwired Platform version 1.5.2 or earlier)
   - Sybase Unwired CacheDB
   - Sybase Unwired ClusterDB service (available with Unwired Platform version 2.0 or later new installations)
   - Sybase Unwired LogDataDB service (available with Unwired Platform version 2.0 or later new installations)
3. Install the upgrade on the passive node, with all Unwired Platform resources online on the active node.

   Both data tier and Unwired Server are upgraded on the passive node.

   Note: Do not begin upgrading any secondary servers until the primary server has completely restarted all its services after being upgraded.

   a) Extract the patch archive file to a temporary location.
   b) Logging in with administrator privileges, open a Windows Command Prompt (cmd.exe) and change to the folder where you extracted the upgrade files.
   c) Run `installebf.bat`.

      The script runs an initial validation check on the files extracted from the upgrade package. If files are corrupted, delete the corrupted archive and download another copy of the upgrade package.

      If the script detects environment issues, you see an error message and the upgrade process stops. Follow the instructions provided and restart the upgrade. If you cannot correct the issue, contact Sybase Technical Support.

      The script does not detect locked files in the installation directory tree. The error message that appears when the script encounters locked files varies.

      If the installer stops with an unexpected error message, look for locked files in the installation directory tree. Unlock any locked files you find and re-run the upgrade script.
   d) Press **Y**, then **Enter** to install the upgrade.
   e) When the upgrade finishes, review the text on your screen for error messages.

      On successful completion of the upgrade, the services for the upgraded server are started, even if you have specified that services for the server are to be manually started.

      Note: After upgrading the data tier, the first Unwired Server may take as long as an hour to restart.

   f) Before proceeding, verify that the upgraded server is started.
4. Validate the upgrade on this passive node by starting the Unwired Platform services.

5. For the data tier server, make the passive node active.
   a) Launch Microsoft Cluster Administrator.
   b) Fail over the active node to the passive node.

   The node that was passive is now active.
6. Install the data tier upgrade on the new passive node.

   Both the data tier and Unwired Server are upgraded.
   a) Extract the patch archive file to a temporary location.
   b) Logging in with administrator privileges, open a Windows Command Prompt (cmd.exe) and change to the folder where you extracted the upgrade files.
   c) Run `installebf.bat`.

      The script runs an initial validation check on the files extracted from the upgrade package. If files are corrupted, delete the corrupted archive and download another copy of the upgrade package.

      If the script detects environment issues, you see an error message and the upgrade process stops. Follow the instructions provided and restart the upgrade. If you cannot correct the issue, contact Sybase Technical Support.

      The script does not detect locked files in the installation directory tree. The error message that appears when the script encounters locked files varies.

      If the installer stops with an unexpected error message, look for locked files in the installation directory tree. Unlock any locked files you find and re-run the upgrade script.
   d) Press **Y**, then **Enter** to install the upgrade.
   e) When the upgrade finishes, review the text on your screen for error messages.

      On successful completion of the upgrade, the services for the upgraded server are started, even if you have specified that services for the server are to be manually started.

      Note: After upgrading the data tier, the first Unwired Server may take as long as an hour to restart.
   f) Before proceeding, verify that the upgraded server is started.

Replacing Existing Sybase Control Center Login Modules with Delegate Login Module

If you have manually configured a login provider to work with Sybase Control Center for authentication, Sybase strongly recommends that you update `csi.properties` and `roles-map.xml` to use the newly added Delegate Login module. The runtime upgrade script automatically replaces PreConfigured User Login module entries when it finds them.

Adding the Delegate Login module simplifies the maintenance of your Sybase Control Center authentication providers because this module delegates authentication and authorization of Sybase Control Center users to the "Admin" security configuration.

See *Making "Admin" Security Configuration Production-Ready* and *Configuration Files* in the *Security* guide.

1. Open *SCC_HOME*\conf\csi.properties and add the Delegate Login module to it, shown in bold in the example below:

```
## This file defines the default CSI configuration

##########################################
## login modules ##
##########################################
## Anonymous Login Module
CSI.loginModule.
1.provider=com.sybase.ua.services.security.anonymous.AnonymousLog
inModule
CSI.loginModule.1.controlFlag=sufficient
CSI.loginModule.1.options.moduleName=Anonymous Login Module
CSI.loginModule.1.options.roles=uaAnonymous

##########################################
## authorizers ##
##########################################
## XML Authorizer
CSI.authorizer.
1.provider=com.sybase.ua.services.security.xml.XMLAuthorizer

##########################################
## attributers ##
##########################################
## XML Attributer
CSI.attributer.
1.provider=com.sybase.ua.services.security.xml.XMLAttributer


## SUP PreConfiguredUser and Delegation Login Modules
## The number "2" is file content dependent and must be derived
## by the user based on the quantity of other modules configured
## in the same file. The "modulename" must be unique.
## A controlFlag is defined through Sybase Control Center
## properties for each stacked security provider.
CSI.loginModule.2.options.moduleName=SUP Delegation Login Module
CSI.loginModule.
2.provider=com.sybase.ua.services.security.sup.SUPDelegateLoginMo
dule
CSI.loginModule.2.controlFlag=sufficient
```

2. Open *SCC_HOME*\conf\roles-map.xml, then insert the entries in bold to use only the Delegation Login module:

```
<?xml version="1.0" encoding="UTF-8"?>
<roles-map>
 <uaf-roles>
   <role name="uaAgentAdmin" description="Agent administrator
role" />
   <role name="uaPluginAdmin" description="Plugin administrator
```

```
role" />
   <role name="uaOSAdmin" description="Operation system
administrator role" />
   <role name="uaASEAdmin" description="ASE administrator role" />
   <role name="uaUser" description="User role" />
   <role name="uaGuest" description="Guest role" />
   <role name="uaAnonymous" description="Anonymous role" />
   <role name="sccAdminRole" description="SCC Administrator
Role" />
   <role name="sccOperRole" description="SCC Operator Role" />
   <role name="sccUserRole" description="SCC User Role" />
   <role name="sccGuestRole" description="SCC Guest Role" />
   <role name="jmxDirectAccess" description="JMX Direct Access
Role" />
 </uaf-roles>
 <security-modules>

   ## The "name" attribute value must match the name specified in
the csi.properties file.
<module name="SUP Delegation Login Module">
     <role-mapping modRole="SUP Administrator"
uafRole="uaAnonymous,uaAgentAdmin,uaPluginAdmin,sccAdminRole,sccU
serRole,sccOperRole,sccGuestRole,jmxDirectAccess"/>
     <role-mapping modRole="SUP Domain Administrator"
uafRole="uaAnonymous,uaAgentAdmin,uaPluginAdmin,sccUserRole,sccOp
erRole,sccGuestRole,jmxDirectAccess"/>
   </module>
   <module name="Anonymous Login Module">
     <role-mapping modRole="uaAnonymous" uafRole="uaAnonymous" />
   </module>
 </security-modules>
</roles-map>
```

Upgrading to Sybase Mobile SDK 2.1 ESD #3

Upgrade to Sybase Mobile SDK 2.1 ESD #3 by performing a full installation that coexists with, and retains full backward compatibility with, applications developed in the earlier version of the Mobile SDK.

For instructions on installing this version of Sybase Mobile SDK, and for an explanation of the backward compatibility supported, see the *Installation Guide for Sybase Mobile SDK*.

Adding an Unwired Server Node to a Cluster

Perform a fresh Unwired Server installation using the Unwired Platform version 2.1 installer, then upgrade using the Unwired Platform version 2.1 ESD #3 script.

1. Make sure you have the information required to connect the new node to the existing cluster:

- License file or path to license server for cluster
- UNC path to the file system that houses the cluster's data tier database files
- Windows account under which existing nodes in cluster have been installed
- Information that Unwired Server needs to connect to the cluster's data tier
  - Name of data tier host

    If the data tier is installed in a failover cluster, enter the network name of the cluster resource that acts as a "virtual host" for the data tier services.
  - For each configurable database in the data tier:
    - Database name
    - Port number
    - Login
    - Password
- (Optional) To configure Unwired Platform communication ports, the port numbers to be used for server administration, data change notification, messaging, and replication
- Password for supAdmin user for the cluster

2. Confirm that the installation target system meets all the requirements for Unwired Platform version 2.1 ESD #3.

   See *Supported Hardware and Software*.
3. Verify that you have Administrator privileges on the installation target host.
4. If the JAVA_TOOL_OPTIONS environment variable is set, remove it before you start Unwired Platform installation.

   Check for the JAVA_TOOL_OPTIONS environment variable in both **User variables** and **System variables** panes of the Environment Variables dialog.

   a) Right-click **My Computer** and select **Properties**.
   b) Select the **Advanced** tab, then click **Environment Variables**.
   c) Select JAVA_TOOL_OPTIONS and click **Delete**.
   d) Click **OK** to exit all dialogs.
5. Shut down all Sybase software, Sybase processes, and associated third-party processes running on the installation target host. This includes any database servers, Sybase Control Center, and if Sybase SAP® DOE Connector is present in an upgrade installation, SAP Data Orchestration Engine. To verify that services are stopped, open the Services panel from Windows Control Panel.
6. Start the Sybase Unwired Platform Runtime installer using one of these methods:

   - Insert the Sybase Unwired Platform Runtime installation media.
   - Install from a Web download. Extract all the archive files to the same temporary directory on a local disk, not a shared drive.
   - Install from an image on the network. You must access the installation image from a mapped drive, not a UNC path.

   Browse to the location of the `setup.exe` file, and:

- On Windows 7 and Windows Server 2008 R2, right-click the `setup.exe` file and select Run as Administrator.
- On other, earlier supported versions of Windows, double-click the `setup.exe` file.

7. On the installer welcome page, click **Next**.
8. On the end-user license agreement page, select your country, accept the terms of the license agreement and click **Next**.
9. On the license details page, select your license product edition and license type.

   (Enterprise Server edition only) Enter the number of client licenses.

   Click **Next**.
10. Enter the location of your license file.

    - If you selected **Unserved (local) license**, enter the absolute path to the license file on the installation target host, using only ASCII characters
    - If you selected **Served (remote) license**, enter the host name and TCP port of the license server.

    Click **Next**.

    If you get an error, such as `Failed to check out license`, click **OK**, then click **Back** to confirm the license model (served or unserved), license product edition, and license type.
11. Specify the absolute path to the installation location, which must reside in a local drive on the target host.

    The total length of the path must be 43 characters or less.

    Directory names in the path can contain only:
    - ASCII alphanumeric characters
    - Underscore (_) or hyphen (-) characters
12. On the installation option page, select **Unwired Platform Runtime** and click **Next**.
13. On the installation type page, select **Cluster** and click **Next**.
14. On the cluster installation step page, select **Install an additional server node and connect it to the data tier** and click **Next**.
15. If a page appears, indicating that the installer detected missing third-party software, click:

    - **Next** to install the required software.
    - **Back** to select components to install that do not require the third-party software.
    - **Cancel** to stop the current installation.
16. On the additional installation options panel:

    - (Optional) Select **Configure Unwired Platform communication ports** to change default ports.
    - De select **Set Sybase Unwired Platform services to start automatically** to start services automatically when Windows starts.

If you set Unwired Platform services to start automatically, the new Unwired Server node may fail to sync with the cluster.

- (Optional) Select **Set Sybase Unwired Server name** to specify a name other than the server name designated in the file system.

17. Enter cluster configuration information that the installer needs to access the data tier.

a) Enter the Unwired Platform cluster configuration password.

The cluster configuration password is specified when the data tier is installed.

b) Enter the UNC path to the file system that houses the data tier database files.

The UNC path must use a fully qualified host name or network name (`name.domain`). For example, if the data tier node is installed in `C:\Sybase`, the data folder is `C:\Sybase\UnwiredPlatform\Data`. If the machine is named SUPDT, the UNC path to the data folder `is \\SUPDT.example.com\Data`.

Note: If the data tier is installed in a failover cluster, the UNC path must point to the Network Name resource associated with either a file share resource group or a Client Access Point.

18. (Optional) If you selected **Set Sybase Unwired Server name**, enter a name for this Unwired Server instance. The name must be unique on the network segment.

19. Enter information for the Windows account under which Unwired Server services will run.

The user ID (account name) must:

- Be a Windows domain user ID, preceded by a Windows domain.
- Be a member of the local Administrator's group on the installation target host.
- Have read/write access to the file system that houses the data tier database files.

Note: You must enter identical Windows account information for each node in the Unwired Server cluster.

20. Enter information that Unwired Server needs to connect to the data tier.

a) Enter the name of the data tier host.

b) Enter the following for each configurable database listed:

- **Database Name**
- **Port Number**
- **Login**
- **Password**

In a typical new installation, you need not change any pre-populated value, just enter "sql" as the default password for each database.

In an upgrade installation, if the password fields are filled, do not change them. If they are not filled, enter the current the password for each database.

21. (Optional) If you selected **Configure Unwired Platform communication ports**, change the default port numbers as needed.

- General ports:
  - Server administration
  - Data change notification
- Synchronization ports:
  - Messaging
  - Replication

22. Enter the password for the supAdmin user and click **Next**.

The password must:
- Be at least 8 characters.
- Contain only ASCII alphanumeric characters.

23. On the summary information page, verify the installation features and click **Install**.

24. (Optional) Click **View Release Bulletin**.

25. Click **Finish**.

26. Verify that Unwired Platform services are shut down before proceeding.

27. Run the `installebf.bat` script for the Unwired Platform version 2.1 ESD #3 upgrade.

a) Extract the patch archive file to a temporary location.

b) Logging in with administrator privileges, Open a Windows Command Prompt (cmd.exe)—on Windows 2008 R2, right-click the menu item and select Run as Administrator—and change to the folder where you extracted the upgrade files.

c) Run `installebf.bat`.

The script runs an initial validation check on the files extracted from the upgrade package. If files are corrupted, delete the corrupted archive and download another copy of the upgrade package.

If the script detects environment issues, you see an error message and the upgrade process stops. Follow the instructions provided and restart the upgrade. If you cannot correct the issue, contact Sybase Technical Support.

The script does not detect locked files in the installation directory tree. The error message that appears when the script encounters locked files varies.

If the installer stops with an unexpected error message, look for locked files in the installation directory tree. Unlock any locked files you find and re-run the upgrade script.

d) Press **Y**, then **Enter** to install the upgrade.

e) When the upgrade finishes, review the text on your screen for error messages.

On successful completion of the upgrade, the services for the upgraded server are started, even though you specified that services for the server are to be manually started.

f) Before proceeding, verify that the upgraded server is started.

28. Verify that you can start the following services manually. Open the Services panel from Windows Control Panel verify that services are started:
    - Sybase Control Center
    - Sybase Unwired SampleDB (not applicable to Enterprise Server Edition)
    - Sybase Unwired Server
29. From Windows, select **Start > (All) Programs > Sybase > Sybase Control Center** to log in to Sybase Control Center.
30. Verify that you can connect to the Unwired Server you just installed.
31. Verify that you can connect to the data tier.
32. Verify that you can connect to all the Unwired Server instances you have previously installed.

Installing Online Data Proxy in a Cluster

Unwired Platform version 2.1 ESD #3 supports installation of Online Data Proxy in a cluster.

Installing Online Data Proxy in an Unwired Platform cluster is a two-part process.

1. Install the full version 2.1 Unwired Platform Runtime in a cluster.
   See the version 2.1 *Installation Guide for Runtime*.
2. Upgrade the cluster to Unwired Platform 2.1 ESD #3. The upgrade automatically adds Online Data Proxy functionality; no manual configuration is required.
   See the version 2.1 ESD #3 *README* file.

Note: You cannot migrate an earlier installation of Online Data Proxy into a cluster installation.

Migrating Your Artifacts

After upgrading Unwired Platform Servers, migrate your MBOs, projects, and applications.

Mobile Business Object

You may encounter some issues when migrating mobile business objects (MBOs) to Unwired Platform version 2.1 ESD #3.

Updating the Database Connection Profile

Due to JDBC driver location changes, beginning in Sybase Unwired Platform version 2.1. ESD #3, the existing database connection profile may fail to ping/connect with this exception:

```
java.lang.ClassNotFoundException:
com.sybase.jdbc3.jdbc.SybDriver.
```

If you see this error, change the JDBC driver location. For example, for a SQL Anywhere® connection profile:

1. In Enterprise Explorer, right-click the database connection profile, and select **Properties**.
2. In the left pane, select **Sybase ASA Connection Properties**.
3. Click the upper-right triangle button to invoke the Edit Driver Definition dialog.
4. Select the **Jar List** tab, and select the driver file entry that contains the incorrect JDBC driver file path, then click the **Edit JAR/Zip** button to invoke the Select the file dialog.
5. Click **Look in** to specify the path. For example: `<SUP Installation Root>\MobileSDK22\Unwired_WorkSpace\Eclipse\sybase_workspace\framework\eclipse\plugins\com.sybase.jconnect60_6.0.0.200710200257\lib.`
6. Click **OK** to complete the driver definition location change.
7. Click **OK** to complete the connection profile properties change.

Ensure that Normal MBOs Have Primary Keys

Normal MBOs that have no primary key defined generate a warning message when migrating earlier versions of mobile application projects to Unwired Platform version 2.1 ESD #3. To ensure optimum performance, make sure that normal MBOs have a primary key.

When you migrate a project from an earlier version of Unwired Platform to version 2.1 ESD #3:

- Any normal MBO in the project that does not have a primary key generates a warning message.
- Local business objects without a primary key do not trigger this warning.
- You may ignore the warning on normal MBOs and proceed with the migration. The MBO continues to function as in the earlier version of Unwired Platform, but with performance that is not optimal.
- To ensure optimal performance with migrated MBOs, make sure that all your normal MBOs have a primary key. After adding a primary key, regenerate and redeploy the MBO.

See *Developing a Mobile Business Object* in *Sybase Unwired WorkSpace - Mobile Business Object Development*.

Eliminate Composite Orphans in Native Object API Applications

A composite orphan is an MBO child entity within a composite relationship without a parent entity. For native Object API applications, the upgrade process silently deletes composite orphans from the server cache. Check all your MBOs to ensure that applications run properly after the upgrade.

Unwired Platform version 2.1 ESD #3 no longer allows composite orphans. Depending on the system configuration, the upgrade may physically delete the orphans, or may only logically

delete them, with physical deletion occurring during a subsequent cache purge process. Either way, applications that use MBOs with composite orphans may not behave the same after migration. To avoid problems with existing applications:

- Look for existing MBOs that generate composite orphans. Examine each MBO in Unwired WorkSpace, looking for any instance in which the application loads child entities before the related parent entity.
- For any MBOs you find that generate composite orphans, change the data model in the MBO so that the application always loads parent entities before the related child entities. See the "Composite relationship behavior" row in *Relationship Guidelines and Restrictions* in *Sybase Unwired WorkSpace - Mobile Business Object Development*.
- Test any changes you make to ensure that the application still behaves as expected.
- After you are satisfied that you have eliminated composite orphans from your MBOs, proceed with the upgrade to Unwired Platform version 2.1 ESD #3.

Redeploying Projects That Contain Composite Relationships

After migration if you have mobile application projects that contain composite relationships and the child MBO load operation has no parameter dependencies on its parent, you must redeploy the package from Unwired WorkSpace using Update mode to generate the correct loadGroups for the deployed package.

Child load operations that do not depend upon parent parameters do not always generate orphans. Consider the case where the user wants to load all sales orders and line items created in the New York office. New York may be a load parameter for both parent and child that is bound to a personalization key and results in no orphans, even though the child does not get any load parameters from the parent.

Native Client Object API

Compatibility between versions of the Native Client Object API and Unwired Server.

Native Client Object API	Unwired Server 2.0.x	Unwired Server 2.1	Unwired Server 2.1 ESD #1	Unwired Server 2.1 ESD #2	Unwired Server 2.1 ESD #3
Native Client Object API 2.0.x	Yes	Yes	Yes	Yes	Yes
Native Client Object API 2.1	No	Yes	Yes	Yes	Yes
Native Client Object API 2.1 ESD #1	No	No	Yes	Yes	Yes

Native Client Object API	Unwired Server 2.0.x	Unwired Server 2.1	Unwired Server 2.1 ESD #1	Unwired Server 2.1 ESD #2	Unwired Server 2.1 ESD #3
Native Client Object API 2.1 ESD #2	No	No	No	Yes	Yes
Native Client Object API 2.1 ESD #3	No	No	No	No	Yes

Required Changes for Object API Applications

Change applications to use APIs introduced in SDK version 2.1 ESD #2 and SDK version 2.1 ESD #3. You can migrate client applications built from the previous SDK versions 2.0, 2.0 ESD #1, 2.1, 2.1 ESD #1, or 2.1 ESD #2.

When an existing client application built from a previous SDK version is integrated with the 2.1 ESD #3 SDK and installed on a device, it must be fully functional, with its related server packages deployed prior to performing an upgrade, or deploying a new server package after an upgrade.

Android

No application or project changes are required.

No changes are required to migrate Android applications or projects for SDK version 2.1 ESD #3.

BlackBerry

These changes are required for BlackBerry applications.

Application Changes Required for 2.1 ESD #3

There are no additional application changes required for 2.1 ESD #3. See *Developer Guide: BlackBerry Object API Applications* for information on developing your application.

Application Changes Required for 2.1 ESD #2

Update and rebuild your application:

1. The Application APIs (in the `Application` class) are required for managing application registrations, connections, and context. Rewrite the initialization code in your application to use the Application APIs.
   For information on the `Application` interface, search for *Application APIs* in the Developer Guide for your platform.
2. Callbacks related to application events are contained in a separate `ApplicationCallback` interface. Rewrite your application code to use this interface.

For information on the `ApplicationCallback` interface, search for *Callback and Listener APIs* in the Developer Guide for your platform.

3. Replication-based synchronization clients require two data channels: a data channel for data synchronization, and a messaging channel for sending registration and push notifications to the client. Update your port configuration for both channels. See *Sybase Control Center for Sybase Unwired Platform* > *Administer* > *Unwired Server* > *Server Properties*.
4. To continue using server-initiated synchronization, you must write code for handling notifications. If change notifications are enabled for synchronization groups, you can implement the `onSynchronize` callback method to monitor this condition, and either allow or disallow default background synchronization.

```
public int onSynchronize(ObjectList groups,
SynchronizationContext context)
{
  int status = context.getStatus();
  if (status == SynchronizationStatus.STARTING_ON_NOTIFICATION)
  {
    // There is changes on the synchronization group
    if (busy)
    {
      return SynchronizationAction.CANCEL;
    }
    else
    {
      return SynchronizationAction.CONTINUE;
    }
  }

  // return CONTINUE for all other status
  return SynchronizationAction.CONTINUE;
}
```

5. The new location of the required libraries is `<UnwiredPlatform_InstallDir>\UnwiredPlatform\MobileSDK\ObjectAPI\BB`.

   Rebuild your project as follows:

   a. Modify the BlackBerry project's Java build path to point to the new location for your project.
   b. Replace the old libraries, `sup_client_rim.jar` and `UltraLiteJ.jar`, with these new references:
      - `sup_client2.jar` – from `<UnwiredPlatform_InstallDir>\MobileSDK\ObjectAPI\BB`.
      - `UltraLiteJ1.jar` – from `<UnwiredPlatform_InstallDir>\MobileSDK\ObjectAPI\BB`.
   c. In Windows Explorer, copy the new required files into the BlackBerry simulator directory:
      - `CommonClientLib.cod` – from `<UnwiredPlatform_InstallDir>\MobileSDK\ObjectAPI\BB`.

- `MessagingClientLib.cod` – from `<UnwiredPlatform_InstallDir>\MobileSDK\ObjectAPI\BB`.
- `MocaClientLib.cod.cod` – from `<UnwiredPlatform_InstallDir>\MobileSDK\ObjectAPI\BB`.
- `sup_client2.cod.cod` – from `<UnwiredPlatform_InstallDir>\MobileSDK\ObjectAPI\BB`.
- `UltraLiteJ12.cod` – from `<UnwiredPlatform_InstallDir>\MobileSDK\ObjectAPI\BB`.

d. Delete the old `CommonClient.cod`, `sup_client_rim.cod`, and `UltraLiteJ.cod` files.

iOS

These changes are required for iOS applications.

Application Changes Required for 2.1 ESD #3

The application changes required for 2.1 ESD #3 depend on your migration strategy scenario. See *Migrating iOS Native Custom Applications* on page 52.

For 2.1 ESD #3, use the required version of the iOS SDK. See *Supported Hardware and Software*.

Application Changes Required for 2.1 ESD #2

Update and rebuild your application:

1. The Application APIs (in the`SUPApplication` class) are required for managing application registrations, connections, and context. Rewrite the initialization code in your application to use the Application APIs. Replace references in your application to the Messaging Client API (`SUPMessage` class) with the appropriate use of the Application APIs (`SUPApplication`).

   For information on the `SUPApplication` interface, search for *Application APIs* in the Developer Guide for your platform.
2. Callbacks related to application events are contained in a separate `ApplicationCallback` interface. Rewrite your application code to use this interface.

   For information on the `ApplicationCallback` interface, search for *Callback and Listener APIs* in the Developer Guide for your platform.
3. In addition to the required libraries provided in earlier SDK versions (libMO.a, libclientrt.a, and libSUPObj.a), two new libraries (libsupcore.a, libAfariaSLL.a) are required for all iOS applications. These libraries also require additional Apple-provided frameworks to be included when building object API applications.

   For information on importing required libraries, see *Developer Guide: iOS Object API Applications* > *Developer Task Flow* > *Importing Libraries and Code*.

Migrating iOS Native Custom Applications

Understand the strategies and steps to follow when you transition applications to the current release.

Migration Strategies

Your strategy for transitioning MBS-based iOS applications to the current release depends on your current installation configuration, upgrade plans, and the data model changes in the application to be transitioned. Follow the guidance in the scenario that fits your installation configuration and upgrade plan.

Scenario 1

- Current Installation - 2.1 ESD #2 or earlier MBS client application on 2.1 ESD #2 or earlier Unwired Server
- Upgrade Plan - Upgrade only Unwired Server to the current version, and maintain the existing MBS client application

Your MBS client application should continue to work without error after server upgrade, though some RBS features will not be available for your MBS client application. See *Maintaining MBS Client Applications* on page 54

Scenario 2

- Current Installation - 2.1 ESD #2 or earlier MBS client application on 2.1 ESD #2 or earlier Unwired Server
- Upgrade Plan - Upgrade both Unwired Server and client application to the current version. Upgrade the client application to an RBS-based application.
- No Data Model Changes in the application

Recommended Steps:

1. Instruct application users to submit all pending data to the Unwired Server using the existing MBS client application before you migrate to the new RBS application, and coordinate the upgrade. This is an important step as it will ensure that application users do not lose any modified data during your migration. With MBS, once **submitPending** is invoked, the modified data is wrapped as an operation replay message to be sent as soon as connectivity with the server is available. If the application user does not invoke **submitPending** prior to migration, all of their data changes will be lost once migration begins. For this reason, you will need to instruct the application users to use the appropriate UI control exposed by the MBS application to invoke **submitPending** before you migrate the application.
2. Follow the steps included in *Transitioning MBS Client Applications* on page 54 to convert the MBS application to the new RBS application, creating a different application name for the new RBS application on the device. Include explicit screens/message popups within the application to alert the application user to follow these steps:
   a. Submit all pending data from the MBS client application to the Unwired Server.

   b. Confirm that the pending data has been submitted, delete the MBS application, and then begin using the new RBS application.

   Note: Once the application user acknowledges and confirms that pending data from the old application has been submitted, do not display the popup/screen messages again.

   c. Subscribe and synchronize the new RBS application with the upgraded Unwired Server.

Note: You need to use a different Application Name to avoid an accidental update of the MBS application before the application user has a chance to submit their changes. However, you *can* use the same Application ID for both the new RBS application and for the existing MBS application.

For more in depth steps to transition your MBS client application to RBS, see *Transitioning MBS Client Applications* on page 54

Scenario 3

- Current Installation - 2.1 ESD #2 or earlier MBS client application on 2.1 ESD #2 or earlier Unwired Server
- Upgrade Plan - Upgrade both Unwired Server and client application to the current version. Upgrade the client application to an RBS-based application.
- Data Model Changes in the application or MBO project

Recommended Steps:

1. Instruct application users to submit all pending data to the Unwired Server using the existing MBS application before you migrate to the new RBS-based application, and coordinate the upgrade. This is an important step as it will ensure that application users do not lose any modified data during your migration. With MBS, once **submitPending** is invoked, the modified data is wrapped as an operation replay message to be sent as soon as connectivity with the server is available. If the application user does not invoke **submitPending** prior to migration, all of their data changes will be lost once migration begins. For this reason, you will need to instruct the application users to use the appropriate UI control exposed by the MBS application to invoke **submitPending** before you migrate the application.
2. Deploy the new package with data model changes to the server using a new Application ID. Create a new application connection in the Sybase® Control Center.
3. Follow the steps included in *Transitioning MBS Client Applications to the Current Release* on page 54 to convert the MBS application to the new RBS application, creating a different application name and application id for the new RBS application on the device. Include explicit screens/message popups within the application to alert the user to follow these steps:
   a. Submit all pending data from the MBS client to the Unwired Server.
   b. Confirm that the pending data has been submitted, delete the MBS application, and then begin using the new RBS application.

Note: Once the application user acknowledges and confirms that pending data from the old application has been submitted, do not display the popup/screen messages again.

c. Subscribe and synchronize the new RBS application with the upgraded Unwired Server.

For more in depth steps to transition your MBS client application to RBS, see *Transitioning MBS Applications to the Current Release (2.1.3 ESD #3 or Later)* on page 54

Note: For Scenario 2 and 3, there is no data transitioning solution when migrating MBS applications to RBS applications. After the application is converted to RBS, the application user must synchronize the application with the Unwired Server. The new application will not use the data residing in the device database for the old application so the application user will need to delete the old application from the device. If the old application is not removed from the device, the database for the old application will continue to reside on the device; this may double the space consumed on the device when the new application downloads records to the new database.

Maintaining MBS Client Applications

To continue to use your existing MBS client applications, continue to use an earlier version of the SDK.

When you upgrade your Sybase Mobile SDK, the installation does not overwrite earlier versions of the SDK. Instead, the installation coexists with the earlier version of the SDK, and retains full backward compatibility with applications developed in the earlier version. However, features available in 2.1 ESD #3 or later versions of the SDK may not be available for applications developed in earlier versions of the SDK.

The following replication-based synchronization features are unavailable for messaging-based synchronization applications:

- Asynchronous upload of operation replay results
- Push synchronization APIs for sending change notifications to devices
- Change log APIs to allow a client to retrieve entity changes from the back end

For information on support of earlier SDKs with a 2.1 ESD #3 or later server, see the *Installation Guide for Sybase Mobile SDK > Getting Started > Backward Compatibility*.

For information on messaging-based synchronization applications, see the *Developer Guide: iOS Object API Applications* from 2.1 ESD #2.

Transitioning MBS Applications to the Current Release (2.1 ESD #3 or Later)

(Not applicable to DOE based applications) iOS applications built with earlier versions of the SDK use messaging-based synchronization (MBS) for data delivery. Applications built using

SDK version 2.1 ESD #3 or later use replication-based synchronization (RBS) for data delivery to reduce synchronization time.

This task flow shows you how to transition your messaging-based application to the current release as a replication-based application. The tasks include setting up the project, updating the application, and testing the application.

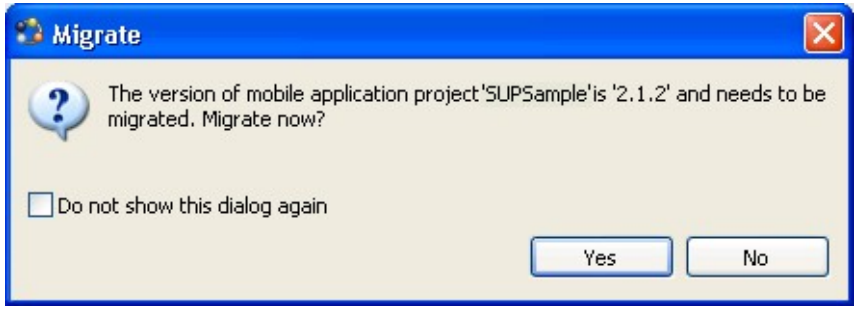
Note: The code samples in this task flow are from the SUP101 project from the *Tutorial: iOS Object API Application Development*.

Migrating the Project and Generating Code

Migrate the existing project to the current version of Sybase Mobile SDK, and generate new RBS object API code.

Important: Upgrade to the current version of Sybase Mobile SDK prior to migrating your project.

1. Export the existing mobile application project from the earlier version of Unwired WorkSpace.
2. In the current version of tooling-name, import your existing application project.
3. Right-click the project and select **Open in Diagram Editor**.
4. Select **Yes** to migrate the project to the current version of the SDK.



5. Right-click the project and select **Generate Code** to generate code that supports replication-based synchronization.

For more information on code generation options, see *Developer Guide: iOS Object API Applications > Developer Task Flow for Object API Applications > Generating Objective-C Object API Code*.

Setting Up the Xcode Project

Set up the Xcode project with the generated code and libraries required in SDK version 2.1 ESD #3.

Important: Install the Xcode version required for SDK version 2.1 ESD #3 prior to setting up the Xcode project. See *Supported Hardware and Software*

1. In the Xcode project, open your existing application.
2. Remove the existing generated code and add the new generated code.
   To remove the existing generated code:
   a. In the Xcode tree view, right-click the `Generated Code` folder and select **Delete**.
   b. In the confirmation dialog, select **Delete**.
   c. In Finder, go to the Xcode project folder. Delete the empty `Generate Code` physical folder to ensure that the new generated code for SDK 2.1 ESD #3 gets imported correctly by Xcode.
3. Remove all of the libraries that you added from the `<Unwired_Platform_Install_Dir>\MobileSDK\ObjectAPI\iOS\Libraries\` folder when you created the application in an earlier version of the SDK.
4. Add all of the libraries from the `<Unwired_Platform_Install_Dir>\MobileSDK213\ObjectAPI\iOS\RBS\Libraries\` folder in SDK version 2.1 ESD #3.
5. Remove the existing `\includes` header files and add the new ones from `<Unwired_Platform_Install_Dir>\MobileSDK213\ObjectAPI\iOS\RBS\includes\`
   To remove the existing files:
   a. In the Xcode tree view, right-click the `includes` folder and select **Delete**.
   b. In the confirmation dialog, select **Delete**.
   c. In Finder, go to the Xcode project folder. Delete the empty `includes` physical folder to ensure that the new generated code for SDK 2.1 ESD #3 gets imported correctly by Xcode.

Making Changes to Application Registration

This task is not required if your application is built with SDK version 2.1 ESD #2. Make changes to the application to allow it to register as required in 2.1 ESD #3.

1. The Application APIs (in the `Application` or `SUPApplication` class) are required for managing application registrations, connections, and context. Rewrite the initialization code in your application to use the Application APIs. For information on the `Application` interface, search for *Application APIs* in the Developer Guide for your platform.
   For iOS applications, the Messaging Client API has been removed. Replace references in your application to the Messaging Client API (`SUPMessage` class) with the appropriate use of the Application APIs (`SUPApplication`).
2. Callbacks related to application events are now contained in a separate `ApplicationCallback` interface. Rewrite your application code to use this interface. For information on the `ApplicationCallback` interface, search for *Callback and Listener APIs* in the Developer Guide for your platform.
3. Complete application registration through an automatic or manual process. See the *Application and User Management Overview* topic group in *System Administration*.

Use the `SUPApplicationCallback` APIs to check that the application successfully registered and the messaging client connection is established.

Making Changes to Application Initialization

Make changes to the application to allow it to initialize as required in 2.1 ESD #3.

1. Locate the following initial synchronization code from the *Tutorial: iOS Object API Application Development* in 2.1 ESD #2.

   ```
   [SUP101SUP101DB beginOnlineLogin:supuser password:suppass];
   ```

   In an MBS application, subscribe caused data to be pushed to the client from the server. In your RBS application, you must invoke `synchronize` (synchronous replay) or `beginSynchronize` (asynchronous replay).

   An RBS application can synchronize through either a synchronous or asynchronous process. By default, asynchronous replay is enabled. When asynchronous replay is enabled, the server returns after processing the operation replay records that are uploaded to the server in the synchronization session When asynchronous replay is disabled, the server returns after processing the operation replay records that are uploaded to the server, then downloads the results of those operation replay records to the device.
2. Set the login credentials for the database synchronization.

   ```
   SUPConnectionProfile *sp = [SUPSampleSUPSampleDB
   getSynchronizationProfile];
   [sp setUser:@"supAdmin"];
   [sp setPassword:@"supPwd"];
   ```
3. The generated code already sets the required connection properties for database synchronization. If you are using manual registration and change the default connection properties (for example, if you are connecting through a Relay Server), then reset them as follows:

   ```
   [sp setServerName:@"relayservername.com"];
   [sp setNetworkProtocol:@"networkProtocol"];
   [sp setPortNumber:portNumber];
   [sp
   setNetworkStreamParams:@"trusted_certificates=certificateName.com
   ;compression=zlib;url_suffix=farmIDForRBS"];
   ```
4. Replace the code with code for either asynchronous or synchronous synchronization. For synchronous synchronization:

   ```
   [sp setAsyncReplay:NO];

   @try {
     [SUP101SUP101DB synchronize];
   }
   @catch (NSException *exception) {
     MBOLogError(@"%@: %@", [exception name], [exception reason]);
   }
   ```

   If the `AsyncReplay` flag is turned off, the client object API calls the `onSynchronize` callback method with an SUPSynchronizationStatus_ FINISHING status after the `synchronize`.

Note: In MBS, the generated operation is automatically sent to the Unwired Server. In your RBS applications, you must instead invoke the `synchronize` method to send the record to the Unwired Server. The `synchronize` method is a blocking call.

For asynchronous synchronization:

```
[sp setAsyncReplay:YES];
@try {
[SUP101SUP101DB synchronize];
// or using beginSynchronize to create a background
synchronization request.
//   SUP101SUP101DB beginSynchronize];

}
@catch (NSException *exception) {
  MBOLogError(@"%@: %@", [exception name], [exception reason]);
}
```

If the `AsyncReplay` flag is turned on, the client object API calls the `onSynchronize` callback method with an SUPSynchronizationStatus_ASYNC_REPLAY_UPLOADED status after the `synchronize`, followed by an SUPSynchronizationStatus_FINISHING status.

Note: Control returns immediately, without the replay results synchronized to the client. The `beginSynchronize` method is a nonblocking call. You can use this method from the `SUPDefaultCallbackHandler` to get callbacks once the replay completion notification is received.

```
-  (SUPSynchronizationActionType)onSynchronize:
(SUPObjectList*)syncGroupList withContext:
(SUPSynchronizationContext *)context
```

The above code examples synchronize the default group. Alternatively, you can synchronize based on the synchronization group the MBO belongs to. Replace the code `[MBO getSynchronizationGroup]` in your application with:

```
NSString *mbo_sg = [mbo metaData].synchronizationGroup;
[db synchronize:mbo_sg];
```

The following methods in the registered callback handlers of the database class are called when the replay results are received and processed by the client framework.

- `- (void)onReplayFailure:(id)entityObject`
- `- (void)onReplaySuccess:(id)entityObject`

When you disable `AsyncReplay` in the synchronization profile, the database class `synchronize` is a blocked call. When the method returns, the server data has been synchronized in the database. You no longer need to include a wait period in the code. Remove this code from your application:

```
while ([SUP101SUP101DB hasPendingOperations])
{
```

```
  [NSThread sleepForTimeInterval:1];
}
```

Connecting Through a Relay Server

An iOS RBS client that connects through a Relay Server needs two different farm IDs: one for a messaging client connection to register the application, and the RBS connection for database synchronization.

In your iOS application, set up the messaging client and database connection through Relay Server. Note that, in most cases, the application template already contains settings for the RBS connection so you do not need to set any properties. The settings from the template are downloaded to the client after registration is completed. However, it may be necessary in a development environment to directly manipulate the settings.

1. To set up a messaging client connection, use:

```
SUPApplication * app = [SUPApplication getInstance];

// should be same as application id from SCC
[app setApplicationIdentifier:@"appId"];
SUPConnectionProperties* props = app.connectionProperties;
[props setServerName:serverName];
[props setPortNumber:80]; // or 443 for HTTPS
[props setNetworkprotocol:@"http"]; // or https for secure
connection
[props setUrlSuffix:@""];
[props setFarmId:@"farmIDMBS"];
SUPLoginCredentials* login = [SUPLoginCredentials getInstance];
login.username = @"userName"; // same as in Application Connection
login.password = nil;
props.loginCredentials = login;
props.activationCode = @"123"; // same as in Application
Connection
props.securityConfiguration = @"admin";
```

2. To set up a database connection:
   - If the application connection template on SCC is configured with all the required Relay Server information, application code only needs to do:

     ```
     SUPConnectionProfile *sp = [SUP101SUP101DB
     getSynchronizationProfile];
     [sp setUser:@"supAdmin"];
     [sp setPassword:@"password"];
     [sp setAsyncReplay:NO];
     ```

   - Otherwise, application code needs to fill all the Relay Server information before doing data synchronization:

     ```
     SUPConnectionProfile *sp = [SUP101SUP101DB
     getSynchronizationProfile];
     [sp setUser:@"supAdmin"];
     [sp setPassword:@"password"];
     [sp setAsyncReplay:NO];

     [sp setServerName:@"relayServerHostName"];
     [sp setPortNumber:443]; // or 80 for http
     ```

```
[sp setNetworkProtocol:@"https"];
// certificateName: this should come from the relay server and
should be
// included in the Resource folder of the XCode project
[sp
setNetworkStreamParams:@"trusted_certificates=certificateName;
compression=zlib;url_suffix=urlSuffixRBS"];
```

Note: `urlSuffixRBS` needs to match the exact string of Relay Server RBS url_suffix configuration.

The above code should be done before doing any data synchronization (including subscribe/ onlineLogin).

Setting Up Callbacks

Update your application to use callbacks from SDK version 2.1 ESD #3.

All callback methods are included in the `SUPCallbackHandler` protocol, and you must implement them in any class that directly implements the protocol without subclassing the default implementation in SUPDefaultCallbackHandler.

1. If you have directly implemented the `SUPCallbackHandler` protocol, you must implement all methods. In replication-based synchronization, there are several methods in the protocol that are specific to messaging-based synchronization, and will never be called.

   If you have created your callback handler as a subclass of `SUPDefaultCallbackHandler`, as Sybase recommends, you can safely remove the following messaging-based synchronization callbacks, as the `SUPDefaultCallbackHandler` has empty implementations of all the required methods.

   - `beforeImport`, `onImport`, and `onImportSuccess`
   - `onLoginSuccess`
   - `onSubscribeFailure`, and `onSubscribeSuccess`
   - `onSuspendSubscriptionFailure`, and `onSuspendSubscriptionSuccess`
   - `onResumeSubscriptionFailure`, and `onResumeSubscriptionSuccess`
   - `onUnsubscribeFailure`, and `onUnsubscribeSuccess`
   - `onMessageException`
   - `onTransactionCommit`, and `onTransactionRollback`
   - `onRecoverFailure`, and `onRecoverSuccess`
   - `onSubscriptionEnd`

   For a complete list of callbacks you can implement in your application, see *Developer Guide: iOS Object API Applications > Client Object API Usage > Callback and Listener APIs*.

2. If your application uses `SUPApplicationCallback`, update it to use these methods:
   Old method:

```
- (void)onConnectionStatusChanged:(SUPInt)connectionStatus :
(SUPInt)errorCode :(SUPNullableString)errorMessage;
```

   New method:

```
- (void)onConnectionStatusChanged:
(SUPConnectionStatusType)connectionStatus :(int32_t)errorCode :
(NSString*)errorMessage;
```

   Old method:

```
- (void)onRegistrationStatusChanged:(SUPInt)registrationStatus :
(SUPInt)errorCode :(SUPNullableString)errorMessage;
```

   New method:

```
- (void)onRegistrationStatusChanged:
(SUPRegistrationStatusType)registrationStatus :
(int32_t)errorCode :(NSString*)errorMessage;
```

   Old method:

```
- (void)onDeviceConditionChanged :(SUPInt)condition;
```

   New method:

```
- (void)onDeviceConditionChanged :
(SUPDeviceConditionType)condition;
```

Generating Change Logs

Use the Change Log API to generate change logs that are sent to the client after the synchronization.

In MBS, the application can use the information in the change logs to update its UI tables with new records and deletions. To do in the same in RBS, enable change logs in your application before synchronizing.

```
[SUP101SUP101DB enableChangeLog];
```

This method notifies you of all changes including the initial synchronization records. You may want to set a flag to indicate when the initial synchronization is done so you do not update the UI for all these initial records.

To set a flag, use code similar to this in your callback `onSynchronize` (`isCompleteSynchronize` is an application variable, set to true after the first synchronization is complete):

```
- (SUPSynchronizationActionType)onSynchronize:
(SUPObjectList*)syncGroupList withContext:
(SUPSynchronizationContext*)context
{
  if (context.status == SUPSynchronizationStatus_ERROR)
  {
      MBOLogError(@"onSynchronize failed for context %@ with
```

```
exception %@", context.userContext, [context.exception reason]);
  } else if (context.status == SUPSynchronizationStatus_FINISHING)
  {

      if (self.isCompleteSynchronize)
      {
          // Handle change log
        SUPObjectList *changeLogs = (SUPObjectList *)[SUP101SUP101DB
getChangeLogs:[SUPQuery getInstance]];
          if([changeLogs size] > 0)
          {
              [changeLogs retain];

              // delete these so we don't do updates later on these.
              [SUP101SUP101DB deleteChangeLogs];
              for (id<SUPChangeLog> cl in changeLogs)
              {
                  MBOLogDebug(@"Changelog: %@['%c', %ld]\n",
                              [SUP101SUP101DB getEntityName:[cl
entityType]],
                              [cl operationType], [cl surrogateKey]);

                // If your UI needs to find the actual object you can
                  // convert the entity name to a class.
                  Class entityClass =
NSClassFromString([SUP101SUP101DB  getEntityName:[cl entityType]]);
                  if (entityClass)
                  {
                   // You can either use the surrogate key or change
to the "keyToString" equivalent.
                      NSString *primaryKey = [SUPStringUtil
                                              toString_long:[cl
surrogateKey]];
                      NSString *type = ([cl operationType] == 'D'
                                        ) ? @"delete" : @"update";

                      // Notify your UI with NSNotification...
                  } //entityClass
              }
              [changeLogs release];
          }
      }
  }

  return SUPSynchronizationAction_CONTINUE;
}
```

Creating, Updating, or Deleting Records

In SDK version 2.1 ESD #2 applications, after creating, updating or deleting records, you called the `save` method to save the change to the local database, and called `submitPending` to send the change to the server. In SDK version 2.1 ESD #3 applications, after updating or creating records, you call the `save` and `submitPending` methods, and call `synchronize` to send the changes to the server.

1. In the 2.1 ESD #2 *Tutorial: iOS Object API Application Development*, locate this code:

```
[newCustomer save];
```

```
[newCustomer submitPending];
```

Note: In MBS, the generated operation from `submitPending` is automatically sent to the Unwired Server. In your RBS applications, you must instead invoke the `synchronize` method to send the record to the Unwired Server.

2. Add the following new code. You call `synchronize` to send the update or new record to the server. The call can be either synchronous or asynchronous.

```
@try {
  [SUP101SUP101DB synchronize];
}
@catch (NSException *exception) {
  MBOLogError(@"%@: %@", [exception name], [exception reason]);
}
```

The above code examples synchronize the default group. Alternatively, you can synchronize based on the synchronization group the MBO belongs to.

```
NSString *customer_sg = [customer
metaData].synchronizationGroup;
[db synchronize:customer_sg];
```

Note: Unlike MBS, the `submitPending` method in RBS is a client-side only operation, but is still required before calling the database class's `synchronize` method, which sends the changes to the server.

Testing the Application

After you have transitioned your application to SDK version 2.1 ESD #3, test the application to ensure that it can establish messaging and database connections to the Unwired Server, perform an initial synchronization, and update the database.

Note: There is no data-transitioning solution. The data residing in the old device database is not used after the application is converted to RBS. The application users should submit all pending data to the Unwired Server using the existing MBS client application before the migration to the new 2.1 ESD #3 RBS application. See *Migration Strategies for 2.1 ESD #3* in *Migrating iOS Native Custom Applications* on page 52 After all the pending changes are synchronized to the Unwired Server, the application user needs to remove the old application and/or the older existing database on the device. If the old application is not removed from the device, the database for the old application will continue to reside on the device; this may double the space consumed on the device when the new application downloads records to the new database.

Start and test the client application:

1. Verify that no exceptions have been received from the code that subscribes to the database. If an exception has been received, check the connection profile.

If no exception has been received, you have successfully established the connection to the database.

2. Verify that no exceptions have been received from the code that performs initial synchronization. If an exception has been received, check for any server-side issues in the server log. Also ensure that there is no incompatibility in versions between the deployed package on the server and the generated code.

   If no exception has been received, you have successfully performed an initial synchronization.
3. Verify that no exceptions have been received from the code that creates or updates a record. Also verify that you can view the update on the server.

Windows and Windows Mobile

These changes are required for Windows and Windows Mobile applications.

Application Changes Required for 2.1 ESD #3

In 2.1 ESD#3, there are two new required libraries for Windows clients.

Rebuild your project to include additional references to the new libraries:

1. Add the following new libraries as items in the Visual Studio project. Set the "Build Action" to **Content** and "Copy to Output Directory" to **Copy always**.
   - For Windows:
     - libeay32.dll – from `<UnwiredPlatform_InstallDir>\MobileSDK\ObjectAPI\Win32\`.
     - ssleay32.dll – from `<UnwiredPlatform_InstallDir>\MobileSDK\ObjectAPI\Win32\`.
2. Verify that you have added all required references to your client projects as described in *Developer Guide: Windows and Windows Mobile Object API Applications* > *Development Task Flow for Object API Applications* > *Creating a Project* > *Adding References to a Mobile Application Project*.

See *Developer Guide: Windows and Windows Mobile Object API Applications* for information on developing your application.

Application Changes Required for 2.1 ESD #2

Update and rebuild your application:

1. The Application APIs (in the `Application` class) are required for managing application registrations, connections, and context. Rewrite the initialization code in your application to use the Application APIs.

   For information on the `Application` interface, search for *Application APIs* in the Developer Guide for your platform.
2. Callbacks related to application events are now contained in a separate `ApplicationCallback` interface. Rewrite your application code to use this interface.

For information on the `ApplicationCallback` interface, search for *Callback and Listener APIs* in the Developer Guide for your platform.

3. Replication-based synchronization clients require two data channels: a data channel for data synchronization, and a messaging channel for sending registration and push notifications to the client. Update your port configuration for both channels. See *Sybase Control Center for Sybase Unwired Platform* > *Administer* > *Unwired Server* > *Server Properties*.
4. To continue using server-initiated synchronization, you must write code for handling notifications. If change notifications are enabled for synchronization groups, you can implement the `onSynchronize` callback method to monitor this condition, and either allow or disallow default background synchronization.

```
public int OnSynchronize(GenericList<ISynchronizationGroup>
groups, SynchronizationContext context)
{
  int status = context.Status;
  if (status == SynchronizationStatus.STARTING_ON_NOTIFICATION)
  {
    // There is changes on the synchronization group
    if (busy)
    {
      return SynchronizationAction.CANCEL;
    }
    else
    {
      return SynchronizationAction.CONTINUE;
    }
  }

  // return CONTINUE for all other status
  return SynchronizationAction.CONTINUE;
}
```

5. In 2.1 ESD #2, the new location of the required libraries is `<UnwiredPlatform_InstallDir>\UnwiredPlatform\MobileSDK\ObjectAPI`.

   Rebuild your project as follows:

   a. Reset the references of the following libraries for the appropriate device platform in the Visual Studio project according to the new location:
      - For Windows Mobile:
        - sup-client.dll – from `<UnwiredPlatform_InstallDir>\MobileSDK\ObjectAPI\WM`.
        - iAnywhere.Data.UltraLite.dll – from `<UnwiredPlatform_InstallDir>\MobileSDK\ObjectAPI\WM\Ultralite`.
        - iAnywhere.Data.UltraLite.resources.dll (several languages are supported) – from `<UnwiredPlatform_InstallDir>\MobileSDK\ObjectAPI\WM\Ultralite\<language>`.

- For Windows:
  - sup-client.dll – from <UnwiredPlatform_InstallDir>\MobileSDK\ObjectAPI\Win32.
  - iAnywhere.Data.UltraLite.dll – from <UnwiredPlatform_InstallDir>\MobileSDK\ObjectAPI\Win32\Ultralite.
  - iAnywhere.Data.UltraLite.resources.dll (several languages are supported) – from <UnwiredPlatform_InstallDir>\MobileSDK\ObjectAPI\Win32\Ultralite\<language>.

b. Remove the following libraries for the appropriate device platform as items in the Visual Studio project. The libraries are no longer required.

- For Windows Mobile:
  - ulnet11.dll
  - mlcrsa11.dll (if HTTPS protocol is used)
  - PUtilTRU.dll
- For Windows:
  - ulnet11.dll
  - mlcrsa11.dll (if HTTPS protocol is used)
  - mlczlib11.dll (if using compression)

c. Add the following libraries for the appropriate device platform as items in the Visual Studio project. Set the "Build Action" to **Content** and "Copy to Output Directory" to **Copy always**.

- For Windows Mobile:
  - ulnet12.dll – from <UnwiredPlatform_InstallDir>\MobileSDK\ObjectAPI\WM\Ultralite.
  - mlcrsa12.dll (if HTTPS protocol is used) – from <UnwiredPlatform_InstallDir>\MobileSDK\ObjectAPI\WM\Ultralite.
  - mlczlib12.dll (if HTTPS protocol is used) – from <UnwiredPlatform_InstallDir>\MobileSDK\ObjectAPI\WM\Ultralite.
  - CMessagingClient.dll – from <UnwiredPlatform_InstallDir>\MobileSDK\ObjectAPI\WM\<DeviceType>. <DeviceType> can be Pocket PC or Smartphone as applicable.
- For Windows:
  - ulnet12.dll – from <UnwiredPlatform_InstallDir>\MobileSDK\ObjectAPI\Win32\Ultralite.
  - mlcrsa12.dll (if HTTPS protocol is used) – from <UnwiredPlatform_InstallDir>\MobileSDK\ObjectAPI\Win32\Ultralite.

- mlczlib12.dll (if using compression) - from `<UnwiredPlatform_InstallDir>\MobileSDK\ObjectAPI\Win32\Ultralite`.
- CMessagingClient.dll – from `<UnwiredPlatform_InstallDir>\MobileSDK\ObjectAPI\Win32`.
- ECTrace.dll – from `<UnwiredPlatform_InstallDir>\MobileSDK\ObjectAPI\Win32`.
- TravelerLib.dll – from `<UnwiredPlatform_InstallDir>\MobileSDK\ObjectAPI\Win32`.
- zlib1.dll – from `<UnwiredPlatform_InstallDir>\MobileSDK\ObjectAPI\Win32`.

Object API Changes from Version 2.1 through Version 2.1 ESD #3

The release bulletin provides a complete summary of changes in the Object API made between Unwired Platform versions 2.1 and 2.1 ESD #3.

The following topics contain a summary of all changes in the Object API. Most of these changes add new functionality and do not affect existing applications.

Object API Changes in SDK Version 2.1 ESD #3

These are the changes in the Object API for SDK 2.1 ESD #3.

iOS Transition to Replication-Based Synchronization

iOS applications built with earlier versions of the SDK use messaging-based synchronization (MBS) for data delivery. Applications built using SDK version 2.1 ESD #3 use replication-based synchronization (RBS) for data delivery, to reduce synchronization time to iOS applications. Consequently, messaging-based synchronization has been removed from the 2.1 ESD #3 SDK. To use your existing messaging-based applications, you can continue to use an earlier version of the Sybase Mobile SDK. Installing Sybase Mobile SDK 2.1.3 does not overwrite an earlier version on the same workstation.

API changes related to the application transition include:

- An RBS application uses either a synchronous or asynchronous synchronization process. By default, asynchronous replay is enabled.
- In an MBS application, when an operation replay record is generated from a call to `submitPending`, the generated operation is automatically sent to Unwired Server. In an RBS application, a `synchronize` method is required to send the record to Unwired Server.
- In an MBS application, a subscription event caused data to be pushed to the client from the server. In an RBS application, a call to `synchronize` (synchronous) or `beginSynchronize` (asynchronous) is required.
- RBS applications do not support MBS callbacks.

Documented in:

- *Release Bulletin*: *Transitioning an Existing Object API Application to 2.1 ESD #3*

Datavault

Table 2. New Datavault Methods

Methods	Platform
New methods in the `Datavault` class: • The `setPasswordPolicy` and `getPasswordPolicy` methods allow you to configure or retrieve password policy settings for a data vault. When you call the `createVault` or `changePassword` APIs, the data vault checks the password for compliance to the password policy. • The `isDefaultPasswordUsed` method checks whether the default password is used by the data vault. • The `changePassword (string, string, string, string)` is an overloaded method that validates that a new password is compatible with the password policy, uses the current password to unlock the vault, and changes the password of the vault to a new password. • The `getDataNames` method provides support for iterating through data. The method returns an array of dataName objects. • The `deleteValue` method provides an intuitive way to delete data from the data vault. Previously, you had to set a value to null to delete it.	Android, BlackBerry, iOS, Windows, and Windows Mobile

Table 3. Changed Datavault Methods

Method, Class, or Property	Platform
Changed methods in the `Datavault` class: • The `createVault` and `unlock` methods now automatically compute a salt value when you pass a salt parameter with a null value, or with an empty string. • The `createVault` and `changePassword` methods now provide an option to automatically generate a vault password, by passing a null or empty string for the `password` or `newPassword` parameters. A default password is computed only if the `defaultPasswordAllowedSetting` in the password policy is set to true (the default is false).	Android, BlackBerry, iOS, Windows, and Windows Mobile

Table 4. Removed Datavault Methods

Methods	Platform
Methods removed from the `Datavault` class: • The `setLockTimeout` and `getLockTimeout` methods for setting and retrieving the length of time a vault remains unlocked have been removed. The control of this setting is now specified by the password policy. • The `setRetryLimit` and `getRetryLimit` methods for setting and retrieving the retry limit of the vault have been removed. The control of this setting is now specified by the password policy.	Android, BlackBerry, iOS, Windows, and Windows Mobile

Documented in:

- *Developer Guide: Android Object API Applications*:
  - *Datavault*
  - *SUPDatavault*
- *Developer Guide: BlackBerry Object API Applications*, and *Developer Guide: Windows and Windows Mobile Object API Applications*: *Datavault*
- *Security*:
  - *Encrypting Device Data*
  - *Securing Sensitive Data On-Device with Data Vault*

Authentication

The following authentication methods are deprecated. Instead of using these methods, Sybase recommends that you store the user name and password in a data vault.

Table 5. Deprecated Authentication Methods

Methods	Platform
Deprecated methods from the generated package database class: • `onlineLogin (string userName, string password)` • `offlineLogin` • `loginToSync`	Android, BlackBerry, Windows Mobile, and Win32

Documented in:

- *Developer Guide: Android Object API Applications*, *Developer Guide: BlackBerry Object API Applications*, and *Developer Guide: Windows and Windows Mobile Object API Applications*, search for:

    - *Connecting to Unwired Server with a Certificate*
    - *Single Sign-On With X.509 Certificate Related Object API*
    - *Logging In*
    - *Check Network Connection Before Login*
    - *Synchronization APIs*
    - *subscribe*
- *Troubleshooting*, search for: *iOS Device Cannot Sync*

Object API Changes in SDK Version 2.1 ESD #2

These are the changes in the Object API in SDK 2.1 ESD #2. If your application is built with an earlier version of the SDK, update your application to include the required or recommended API changes.

Application APIs

Sybase Unwired Platform supports and requires clients to use the Application APIs to manage mobile application registrations, application connections, and context. These APIs are provided in the `Application` (or `SUPApplication`) class.

Documented in

- *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*search for:
    - *Initializing an Application*
    - *Application APIs*

:

ApplicationCallback APIs

Sybase Unwired Platform supports a new `ApplicationCallback` interface that is invoked by events of interest to a mobile application.

Documented in:

- *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*search for:
    - *ApplicationCallback API*

Large Attribute APIs

Sybase Unwired Platform supports new large attribute APIs to allow clients to import large messages containing binary objects (BLOBs) from the server or send a replay message without using excessive memory or throwing exceptions.

These new classes support large attributes:

- `BigBinary` (or `SUPBigBinary`)
- `BigString` (or `SUPBigString`)

Documented in:

- *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*search for:
  - *Large Attribute APIs*

Change Log APIs

(Not available in SDK version 2.1 ESD #2 for iOS; introduced for iOS in SDK version 2.1 ESD #3). New change log APIs allow a client to retrieve entity changes from the back end. If a client application already has a list view constructed, it simply needs to add, modify, or delete entries in the list according to the change logs.

Documented in

- *Developer Guide: Android Object API Applications*
- *Developer Guide: BlackBerry Object API Applications*
- *Developer Guide: Windows and Windows Mobile Object API Applications*search for:
  - *Change Log API*

Asynchronous Operation Replay APIs

(Not available in SDK version 2.1 ESD #2 for iOS; introduced for iOS in SDK version 2.1 ESD #3). The Asynchronous Operation Replay API allows users to submit changed data outside a synchronization session without waiting for EIS execution. When an application calls `submitPending` on an MBO on which a create, update, or delete operation is performed, an operation replay record is created on the device local database.

Documented in:

- *Developer Guide: Android Object API Applications*
- *Developer Guide: BlackBerry Object API Applications*
- *Developer Guide: Windows and Windows Mobile Object API Applications*search for:
  - *Asynchronous Operation Replay*

Encryption Key Management APIs

New methods in the connection profile support managing the encryption key for the local client database.

Documented in:

- *Developer Guide: Android Object API Applications*
- *Developer Guide: BlackBerry Object API Applications*
- *Developer Guide: Windows and Windows Mobile Object API Applications*search for:
  - *ConnectionProfile*
  - *Encrypt the Database*
- *Developer Guide: iOS Object API Applications* search for:
  - *SUPConnectionProfile*
  - *Encrypting the Client Database*

Change Notification APIs for Push Synchronization

(Not available in SDK version 2.1 ESD #2 for iOS; introduced for iOS in SDK version 2.1 ESD #3). Sybase Unwired Platform supports using a messaging channel to send change notifications from the server to the client device. Clients receive device notifications when a data change is detected for any of the MBOs in the synchronization group to which they are subscribed. By default, change notification is disabled. You can enable change notification by setting properties in the synchronization group.

Documented in

- *Developer Guide: Android Object API Applications*
- *Developer Guide: BlackBerry Object API Applications*
- *Developer Guide: Windows and Windows Mobile Object API Applications*search for:
  - *Push Synchronization Applications*

Removed APIs

The Messaging Client APIs (`SUPMessageClient` interface) have been removed. If your application uses the `SUPMessageClient` interface, you must rewrite it to use functionality from `SUPApplication`.

Object API Changes in SDK Version 2.1

Changes in the Object API in SDK 2.1. If your application is built with an earlier version of the SDK, update your application to include the required or recommended API changes.

Object Query Enhancements

The Object API provides increased access to the underlying relational persistence layer. SQL features and functions have been added to the dynamic query method, `ExecuteQuery`.

The additions allow you to:

- (Update) Exclude duplicate entries from result sets using the arbitrary find method `Query.DISTINCT` property.
- (Update) Define a filter condition using an MBO attribute using the new conditions, IN, NOT_IN, EXISTS, and NOT_EXISTS. You can nest queries, as well as use `LIKE` keywords in queries.

- (New) Use aggregate functions.
- (New) Group your results using specific attributes.
- (New) Specify how your results are filtered.
- (New) Combine the results of multiple queries into a single result set.
- (New) Include subqueries in `SELECT` statements.

Documented in *Developer Guide: Android Object API Applications*, *Developer Guide: BlackBerry Object API Applications*, *Developer Guide: iOS Object API Applications*, and *Developer Guide: Windows and Windows Mobile Object API Applications*

- *Query API*

Password Policy Implementation for Existing Data Vault Clients

You can update any client application that currently uses a data vault to take advantage of the new password policy feature.

To implement a password policy for data vaults of existing applications, the platform administrator can choose to update existing application templates, or create and reassign new ones. These policies must then be enforced by modifying the login code used in existing applications. For details, see *Enabling and Configuring a Password Policy* and *Creating a Data Vault that Enforce Password Policy*.

Hybrid Web Container

Compatibility between versions of the Hybrid Web Container and Unwired Server, and Hybrid Web Container and Mobile Workflow applications.

Hybrid Web Container and Unwired Server Compatibility

Client/ Hybrid Web Container	Unwired Server 2.0	Unwired Server 2.0.1	Unwired Server 2.1	Unwired Server 2.1 ESD #1	Unwired Server 2.1 ESD #2	Unwired Server 2.1 ESD #3
Hybrid Web Container 2.0	Yes	Yes	Yes	Yes	Yes	Yes
Hybrid Web Container 2.0.1	No	Yes	Yes	Yes	Yes	Yes
Hybrid Web Container 2.1	No	No	Yes	Yes	Yes	Yes

Client/ Hybrid Web Container	Unwired Server 2.0	Unwired Server 2.0.1	Unwired Server 2.1	Unwired Server 2.1 ESD #1	Unwired Server 2.1 ESD #2	Unwired Server 2.1 ESD #3
Hybrid Web Container 2.1 ESD #1	N/A	N/A	N/A	N/A	N/A	N/A
Hybrid Web Container 2.1 ESD #2	No	No	No	No	Yes	Yes
Hybrid Web Container 2.1 ESD #3	No	No	No	No	Yes	Yes

Note: There was no 2.1 ESD #1 Hybrid Web Container. The 2.1 ESD #1 release shipped with 2.1 clients.

Hybrid Web Container and Mobile Workflow Application Compatibility

Client/ Hybrid Web Container	Mobile Workflow Application 2.0	Mobile Workflow Application 2.0.1	Mobile Workflow Application 2.1	Mobile Workflow Application 2.1 ESD #1	Mobile Workflow Application 2.1 ESD #2	Mobile Workflow Application 2.1 ESD #3
Pre-2.0 Client stack/ Applications	Yes	Yes	No	No	No	No
Hybrid Web Container 2.0	Yes	No	No	N/A	No	No
Hybrid Web Container 2.0.1	Yes	Yes	No	N/A	No	No
Hybrid Web Container 2.1	Yes	Yes	Yes	N/A	No	No
Hybrid Web Container 2.1 ESD #1	N/A	N/A	N/A	N/A	N/A	N/A

Client/ Hybrid Web Container	Mobile Workflow Application 2.0	Mobile Workflow Application 2.0.1	Mobile Workflow Application 2.1	Mobile Workflow Application 2.1 ESD #1	Mobile Workflow Application 2.1 ESD #2	Mobile Workflow Application 2.1 ESD #3
Hybrid Web Container 2.1 ESD #2	Yes	Yes	Yes	N/A	Yes	No
Hybrid Web Container 2.1 ESD #3	Yes	Yes	Yes	N/A	Yes	Yes

Note: There was no 2.1 ESD #1 Hybrid Web Container. The 2.1 ESD #1 release shipped with 2.1 clients.

Migrating Mobile Workflow Projects

Mobile workflow projects work properly after migration without special handling, unless you upgrade to the new workflow client containers.

When upgrading to SDK version 2.1 ESD #3:

- To use the new 2.1 ESD #3 functionality, use the Mobile Workflow Forms Editor to regenerate and repackage mobile workflows developed in earlier versions.
- Any customizations made in earlier versions to the `Custom.js` and `Stylesheet.css` files in the mobile workflow package are maintained and functional after regenerating and repackaging.
- Customizations made in earlier versions of several `.js` files in the mobile workflow package are backed up (to a `.backup` extension) the first time the mobile workflow package is generated in 2.1 ESD #3. They are then replaced with the new 2.1 ESD #3 version. You must manually reapply any changes you made in the earlier version.

  For projects created in version 2.0 or 2.0 ESD #1, the files affected are:
  - `API.js`
  - `Utils.js`
  - `WorkflowMessage.js`

  For projects created in version 2.1, 2.1 ESD #1, or 2.1 ESD #2, the files affected are:
  - `API.js`
  - `Callbacks.js`
  - `Camera.js`
  - `Certificate.js`

  - `ExternalResource.js`
  - `SUPStorage.js`
  - `Timezone.js`
  - `Utils.js`
  - `WorkflowMessage.js`
- For workflows generated in earlier versions, the `custom.js` and `stylesheet.css` files are maintained and functional after regenerating and repackaging. A `custom.js.new` file is created.
- If projects created before 2.1 ESD #2 have controls without labels, after migrating to 2.1 ESD #3, error markers appear. You must manually add any missing labels after migrating to 2.1 ESD #3.

Supported Upgrade Scenarios

Basic guidelines on available options for upgrading from various earlier versions.

When you upgrade Unwired Server to version 2.1 ESD #3, no modifications to the earlier deployed versions of Hybrid Web Containers on the server or clients are necessary for the same functionality to operate correctly.

When you upgrade SDK clients from version 2.0 or 2.0 ESD #1 to version 2.1.x, version 2.0 and 2.0 ESD #1 clients coexist with 2.1.x on BlackBerry and Windows Mobile platforms. Workflow messages and existing version 2.0 mobile workflows remain on the clients. Native applications continue to use version 2.0 or 2.0 ESD #1 binaries and settings.

You can install Mobile Workflow applications of different versions, for example, 2.1 and 2.1 ESD #3, on the same device, but if they have the same AppID and same device ID, they cannot connect to the same Unwired Server. Mobile Workflow applications with different AppIDs installed on the same device can connect to the same Unwired Server.

Android

Guidelines on specific options available for upgrading Android workflow projects from various earlier versions.

If the 2.1 ESD #3 Android Hybrid Web Container is installed on an Android device on which the earlier version of Android Hybrid Web Container is already installed:

- The Hybrid Web Containers coexist on that device or emulator.
- Existing mobile workflow applications and messages remain tied to the earlier version of the container and are not removed.
- New registration is required on Unwired Server and you must reassign mobile workflows to the 2.1 ESD #3 Hybrid Web Container.

BlackBerry

Guidelines on specific options available for upgrading BlackBerry workflow projects from various earlier versions.

If the 2.1 ESD #3 BlackBerry mobile workflow container is installed on a BlackBerry device on which an earlier version of the Hybrid Web Container is already installed, the result depends on the version number of the earlier version.

If the earlier version is 2.0:

- The Hybrid Web Containers coexist on that device or simulator.
- The existing mobile workflow applications and messages remain tied to the earlier version of the container and are not removed.
- A new application connection on Unwired Server is required for the 2.1 ESD #2 container, you must assign mobile workflows to the 2.1 ESD #2 container, and enter the connection settings in the 2.1 ESD #3 application.

If the earlier version is 2.1 or 2.1 ESD #2:

- The version 2.1 ESD #3 Hybrid Web Container replaces the earlier version container.
- Existing mobile workflow applications and messages stay intact and are automatically mapped to the 2.1 ESD #3 container.
- You need not register or reregister application connections on Unwired Server or reassign mobile workflows to the 2.1 ESD #3 container.

iOS

Guidelines on specific options available for upgrading iOS workflow projects from various earlier versions.

If the 2.1 ESD #3 Hybrid Web Container is installed on an iOS device on which the 2.0, 2.0 ESD #1, 2.1, 2.1 ESD#1, or 2.1 ESD#2 iOS Hybrid Web Container is already installed, the results depend on the bundle IDs of the two containers.

If the two containers have the same bundle ID:

- The version 2.1 ESD #3 Hybrid Web Container replaces the 2.0, 2.0 ESD #1, 2.1, 2.1 ESD #1, or 2.1 ESD #2 container.
- Existing mobile workflow applications and messages stay intact and are automatically mapped to the 2.1 ESD #3 container.
- You need not register or reregister application connections on Unwired Server or reassign mobile workflows to the 2.1 ESD #3 container.

If the two containers have different bundle IDs:

- The Hybrid Web Containers coexist on that device or simulator.
- The existing mobile workflow applications and messages remain tied to the earlier version of the container and are not removed.

- A new application connection on Unwired Server is required for the 2.1 ESD #3 container, you must assign mobile workflows to the 2.1 ESD #3 container, and enter the connection settings in the 2.1 ESD #3 application.

Windows Mobile

Guidelines on specific options available for upgrading Windows Mobile workflow projects from various earlier versions.

If the version 2.1 ESD #2 Hybrid Web Container that ships with 2.1 ESD #3 is installed on a Windows Mobile device on which an earlier version of the Hybrid Web Container is already installed, the result depends on the version number of the earlier version.

If the earlier version is 2.0 or 2.0 ESD #1:

- The two Hybrid Web Containers coexist on that Windows Mobile device or emulator.
- The existing mobile workflow applications and messages remain tied to the earlier version Hybrid Web Container and are not removed.
- A new application connection on Unwired Server is required for the 2.1 ESD #2 container, you must assign mobile workflows to the 2.1 ESD #2 container, and enter the connection settings in the 2.1 ESD #2 mobile workflow application.
- An additional Workflow menu item is not added to the menu in the mail application. The existing Workflow menu item in the Messages application maps to the 2.0 or 2.0 ESD #1 container.
- Workflow messages in the inbox open the container version with which they are associated. If coexisting Hybrid Web Containers are assigned to the same workflow and have messages sent to the user, two e-mail messages are sent, one for each registration.

If the earlier version is 2.1:

- The version 2.1 ESD #2 Hybrid Web Container replaces the earlier version container.
- Existing mobile workflow applications and messages stay intact and are automatically mapped to the 2.1 ESD #2 container.
- You need not register or reregister application connections on Unwired Server or reassign mobile workflows to the 2.1 ESD #2 container.

The Windows Mobile Programs folder has icons for accessing 2.0, 2.0 ESD #1, and 2.1 settings. Versions 2.0 and 2.0 ESD #1 icons and names remain. For 2.1.x, two new icons are present: **Workflow Settings** for the 2.1.x settings application and **Sybase Mobile Workflow** for the 2.1.x workflow application.

OData SDK Client

Compatibility between versions of OData SDK clients and Unwired Server.

OData SDK Client and Unwired Server Version Compatibility

OData SDK Client	Unwired Server 2.1	Unwired Server 2.1 ESD #1	Unwired Server 2.1 ESD #2	Unwired Server 2.1 ESD #3
OData SDK Client 2.1	Yes	Yes	Yes	Yes
OData SDK Client 2.1 ESD #1	No	Yes	Yes	Yes
OData SDK Client 2.1 ESD #2	No	Yes	Yes	Yes
OData SDK Client 2.1 ESD #3	No	Yes	Yes	Yes

Administration Client API Changes

There have been changes in the Administration Client API between Unwired Platform version 2.1 ESD #3 and earlier versions.

The following table shows the Administration Client APIs that have been changed in Unwired Platform version 2.1 ESD #2 and earilier, and the corresponding api for version 2.1 ESD #3.

Table 6. Administration Client API Changes

Version 2.1 ESD #2 and Earlier API	Corresponding Version 2.1 ESD#3 API
`List<PropertyDefinition2VO> getPropertyDefinitions()`	No change.
`void addDeviceTemplate(String templateName, String templateDescription, List<PropertyItemVO> propertyItems)`	Corresponding `SUPApplication` API: `void createApplicationConnectionTemplate(String name, String description, AppConnectionSettingVO settings)`

Version 2.1 ESD #2 and Earlier API	Corresponding Version 2.1 ESD#3 API
`List<DeviceInfoVO> registerMBSDevicesByTemplate(String templateName, List<RegistrationRequestVO> registrationRequests, List<PropertyItemVO> settings)`	Corresponding `SUPApplication` API: `Collection<Integer> registerApplicationConnections(String templateName, Collection<AppConnectionRegistrationRequestVO> registrationRequests, AppConnectionSettingVO settings)`
`List<DeviceTemplateInfoVO> getDeviceTemplateList()`	Corresponding `SUPApplication` API: `PaginationResult<ApplicationConnectionTemplateVO> getApplicationConnectionTemplates(AppConnectionTemplateFilterSortVO filter, Long offset, Integer count)`
`List<PropertyItemVO> getDeviceTemplateSettings(String templateName)`	Corresponding `SUPApplication` API: `AppConnectionSettingVO getApplicationConnectionTemplateSettings(String templateName)`
`void updateDeviceTemplateSettings(String templateName, List<PropertyItemVO> propertyItems)`	Corresponding `SUPApplication` API: `void updateApplicationConnectionTemplateSettings(String templateName, AppConnectionSettingVO settings)`
`List<DeviceInfoVO> reregisterDevices(List<ReregistrationRequestVO> reregistrationRequests, List<PropertyItemVO> propertyItems)`	Corresponding `SUPApplication` API: `Collection<Integer> reregisterApplicationConnections(Collection<AppConnectionReregistrationRequestVO> reregistrationRequests, AppConnectionSettingVO settings)`
`List<PropertyItemVO> getDeviceSettings(Integer deviceId)`	Corresponding `SUPApplication` API: `AppConnectionSettingVO getApplicationConnectionSettings(Integer numericId)`

Version 2.1 ESD #2 and Earlier API	Corresponding Version 2.1 ESD#3 API
`List<ResponseVO> updateDeviceSettings(List<Integer> deviceIds, List<PropertyItemVO> settings)`	Corresponding `SUPApplication` API: `void updateApplicationConnectionSettings(Collection<Integer> numericIds, AppConnectionSettingVO settings)`
`List<DeviceInfoVO> cloneDeviceRegistration(List<DeviceCloneRequestVO> cloneRequests, List<PropertyItemVO> settings)`	Corresponding `SUPApplication` API: `Collection<Integer> cloneApplicationConnections(Collection<AppConnectionCloneRequestVO> cloneRequests, AppConnectionSettingVO settings)`
`void deleteDeviceTemplates(List<String> templateNames)`	Corresponding `SUPApplication` API: `void deleteApplicationConnectionTemplates(Collection<String> templateNames)`
`List<PropertyItemVO> getDeviceSettings(List<Integer> deviceIds)`	Corresponding `SUPApplication` API: `AppConnectionSettingVO getApplicationConnectionSettings(Integer numericId)` **Note:** The API in `SUPApplication` only supports `get` settings for one application connection.
`void lockDevices(Collection<String> deviceIds)`	Corresponding `SUPApplication` API: `void lockApplicationConnection(Collection<String> applicationConnectionIds)`
`void deleteDevices(Collection<String> deviceIds)`	Corresponding `SUPApplication` API: `void deleteApplicationConnections(Collection<Integer> numericIds)`

Version 2.1 ESD #2 and Earlier API	Corresponding Version 2.1 ESD#3 API
`PaginationResult<DeviceInfoVO> listDevices(DeviceSearchCriteriaVO searchCondition, Long skip, Long take, DeviceSortVO sortInfo)`	Corresponding `SUPApplication` API: `PaginationResult<ApplicationConnectionVO> getApplicationConnections(AppConnectionFilterSortVO filter, Long offset, Integer count)` See *AppConnectionFilterSortVO* Javadoc for more details.
`Collection<DeviceInfoVO> listDevicesByUser(String userName)`	Corresponding `SUPApplication` API: `PaginationResult<ApplicationConnectionVO> getApplicationConnections(AppConnectionFilterSortVO filter, Long offset, Integer count)` See *AppConnectionFilterSortVO* Javadoc for more details.
`PaginationResult<USerInfoVO> listUsers(UserSearchCriteriaVO searchCondition, Long skip, Long take, USerSortVO sortInfo)`	Corresponding `SUPApplication` API: `PaginationResult<ApplicationUSerVO> getApplicationUsers(ApplicationUSerFilterSortVO filter, java.lang.Long offset, java.lang.Integer count)` See *AppConnectionFilterSortVO* Javadoc for more details.
`Collection<UserInfoVO> listUsersByDevice(String deviceId)`	Corresponding `SUPApplication` API: `PaginationResult<ApplicationConnectionVO> getApplicationConnections(AppConnectionFilterSortVO filter, Long offset, Integer count)` See *AppConnectionFilterSortVO* Javadoc for more details.
`void unlockDevices(Collection<String> deviceIds)`	Corresponding `SUPApplication` API: `void unlockApplicationConnections(Collection<String> applicationConnectionIds)`

Version 2.1 ESD #2 and Earlier API	Corresponding Version 2.1 ESD#3 API
`void deleteUsers(List<String> users)`	Corresponding `SUPApplication` API: `void deleteApplicationUsers(java.util.Collection<ApplicationUserVO> users)`
`DeviceInfoVO upgradeRbsDeviceAsMbsDevice(String deviceId, String templateName, RegistrationRequestVO registrationRequest, List<PropertyItemVO> settings)`	Deprecated in Unwired Platform version 2.1.
`void purgeDevices(Integer days, Boolean synchronous)`	Corresponding `SUPApplication` API: `deleteApplicationConnections(java.util.Collection<java.lang.Integer> numericIds)`
`void purgeUsers(String securityConfiguration, Integer days Boolean synchronous)`	Corresponding `SUPApplication` API: `void deleteApplicationUsers(java.util.Collection<ApplicationUserVO> users)`
`PaginationResult<DeviceInfoVO> searchDevicebyMBSProperties(MBSPropertiesSearchCriteriaVO searchCondition, Integer skip, Integer take, MBSPropertiesSortVO sortInfo)`	Corresponding `SUPApplication` API: `PaginationResult<ApplicationConnectionVO> getApplicationConnections(AppConnectionFilterSortVO filter, Long offset, Integer count)` See *AppConnectionFilterSortVO* Javadoc for more details.
`void registerDeviceUser(String userName, List<String> deviceIds, String domain, String pkgName)`	Corresponding `SUPApplication` API: `void registerPackageUsers(String userName, Collection<String>appConnIds, String domain, String pkgName)`

Changed or Deprecated Features

Some Unwired Platform behavior has changed or been deprecated.

CR #	Description
705242	**Messaging-based synchronization (MBS) on iOS is removed in SDK version 2.1 ESD #3.** **Workaround**: Transition iOS applications to use replication-based synchronization (RBS) as soon as you can. If you do not want to transition at this time, you can maintain your iOS applications in an earlier version of the SDK, and deploy them to Unwired Server version 2.1 ESD #3.
703707	**onSynchronize callback handler is now called by the synchronize method.** In 2.1 ESD #3, the `onSynchronize` callback handler is now called by the `synchronize` method. When migrating your application to 2.1 ESD #3, if you register a CallbackHandler, ensure that your application handles `onSynchronize` for the `synchronize` method.
CSN 0120061532 3869546 2011	**OData BlackBerry deprecations for API**. `getSignedCertificateFromStore()` is deprecated and the pop-up screen has been removed from the new API Certificate Selection.
690900	**Bulk download methods deprecated for APIs.** Bulk download methods have been deprecated in Windows Mobile and Win32 APIs. Sybase recommends that you update the initial loading model for newly deployed applications. Existing customers will be supported, but only until the functionality is fully removed. See the corresponding *Mobile Data Models* guide for alternate implementations.

Known Issues for Unwired Platform Runtime 2.1 ESD #3

Learn about known issues and apply workarounds for Unwired Platform runtime components.

Known Issues with Installing, Uninstalling, and Upgrading

Learn about known runtime installation, uninstallation, and upgrade issues and workarounds.

See the *Sybase Unwired Platform Installation Guide for Runtime* for upgrade and migration information regarding runtime, and *Sybase Unwired Platform Installation Guide for Sybase Mobile SDK* for installation information regarding the SDK.

Issue Number	Description

RTC-270	**Test package deployment is skipped when upgrading to Unwired Server 2.1.3**
	When upgrading to Unwired Server 2.1.3, the installer prompts you for an admin ID and password at the end of the upgrade. If the credentials you provide are deemed invalid, the following message is displayed:
	`The Unwired Platform admin login information provided are not valid.  Would you like to try again?`
	`Press Y to re-enter Unwired Platform admin login information.  Any other key to exit`
	If you perform a silent upgrade, you do not have the chance to re-enter different credentials if they are invalid. In this case, the message reads:
	`The admin credential provided was not valid so cannot verify that the server is running properly.  Manually verify that the server has started.  Exiting...`
	Workaround: Ignore the message. You can perform a manual package deployment if you wish to be sure that the upgraded server is function properly. See *Creating new users* in *SQL Anywhere Server - Database Administration*, available at *http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.12.0.0/dbadmin/umannew.html*.

CR 713207	**New Unwired Server node fails to sync with cluster.** After upgrading a cluster to 2.1 ESD #3, an new Unwired Server node that is added fails to sync with the cluster. **Workaround 1**: You can avoid this problem by setting Unwired Server services to start manually when you install the 2.1 server. See *Adding an Unwired Server Node to a Cluster* on page 41. **Workaround 2**: If you have already installed the Unwired Server version 2.1, and upgraded to 2.1 ESD #3, forcing a cluster sync will correct the problem. • Ensure you have logged in to Sybase Control Center as super administrator, and authenticated to the cluster correctly. • In the left navigation pane of Sybase Control Center, select **Security**. • In the right administration pane, click **New**. • Enter a name for the security configuration and click **OK**.

Issue Number	Description
CR 709102	**Unwired Platform 2.1 Installer** When Runtime 2.1 installer is launched on the system with Mobile SDK 2.1 ESD #3 or newer installed already, user will see “null” as the installation destination path which is not changeable. **Workaround 1**: Install 2.1 runtime before Mobile SDK. **Workaround 2**: If Mobile SDK is already installed, do the following: • **32-bit**: Prior to installing 2.1, in the Windows Registry, rename HKLM\Software\Sybase\MobileSDK\2.1.3 to HKLM\Software\Sybase\MobileSDK\2.1. Once the 2.1 runtime install is complete, undo the rename. • **64-bit**: Prior to installing 2.1, in the Windows Registry, rename HKLM\Software\Wow6432Node\Sybase\MobileSDK\2.1.3 to HKLM\Software\Wow6432Node\Sybase\MobileSDK\2.1 in the Windows Registry. Once the 2.1 runtime install is complete, undo the rename.
CR 707703	**2.1 ESD #3 Upgrade** On the first Unwired Server node, upgrades to 2.1 ESD #3 may take up to an hour in a clustered environment with a Windows 2008 R2 data tier server. **Workaround**: None. The upgrade will complete successfully.
CR 696536	**Cluster upgrade successful from 2.0 ESD #1 to 2.1 ESD #3, but device client synchronization fails.** After successfully upgrading from 2.0 ESD #1 to 2.1 ESD #3 in a cluster, device client synchronization fails. This is due to a host name change that was introduced in Sybase Unwired Platform 2.1 for the data tier host; the **serverName** property for the `messagingdb` file is set to the host name of the first server node, and not to the data tier host. This is not an issue for single-node installations. **Workaround**: 1. Navigate to `<UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\UnwiredServer\Repository\Instance\com\sybase\djc\sql\DataSource`. 2. Use a text editor to open `messagingdb.properties`. 3. Fix the `serverName` value to use the fully qualified host name or network name (`name.domain`) from the UNC path to the file system that houses the data tier database files, and not to the primary Unwired Server name. For example, if `\\name.domain.com\Data` was the UNC path to the file system that houses the data tier database files, update the `serverName` value to "`name.domain.com`." 4. Save the file, and repeat the process for the second node. 5. Restart the Sybase Unwired Platform nodes.

Issue Number	Description
CR 694214	**Silent installation fails on shared host Microsoft cluster** If you attempt to perform a silent installation for a shared-host Microsoft cluster scenario, you can successfully install the data tier component, but when you try to install Unwired Server on the same host, the silent installation fails with: `The data path must be in UNC format when installing a cluster node.` **Workaround:** Run the installer in graphical mode.

CR 692712	**When upgrading to Sybase Unwired Platform 2.1 ESD #3, you receive an error message:**
	`"***Error!!! Error!!!***`
	`The Messaging installer completed with the following errors. These errors will have to`
	`be addressed before the services become functional.`
	`Please refer to documentation for recovery process.`
	`Database upgrade failed."`
	This error happens when the messaging server database upgrade does not complete. The most likely reason for this is that the messaging server database retained connections to a table while the upgrade was running. You can get more detailed information from the trace file in `<UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\MessagingServer\Trace\DBUpgrader`.
	Workaround:
	1. Stop all Sybase Unwired Server services in all application nodes.
	2. Restart the database services in the data tier node to ensure the messaging database has no open connections.
	3. From the primary application nodes, open a Windows Command Window and navigate to `<UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\MessagingServer\Bin`.
	4. Run the command **dbupgrader –v current** and make sure no error messages appear.
	5. Run the command **tracemigrator –i** and ensure no error messages appear. You can safely disregard a message that says no action is taken because the migration is already done.
	6. Verify that the Sybase Unwired Server service in the application node can start.

Issue Number	Description
CR 689287	**2.0 or 2.0 ESD #1 MBS clients and Hybrid Web Container clients may not be able to connect to 2.1 ESD #3 server.** If 2.0 or 2.0 ESD #1 clients have not connected through an activated registration, and you upgrade to 2.1 ESD #3, those clients may not be able to connect to the 2.1 ESD #3 server. **Workaround**: Implement a server registration that uses an empty application ID, and use manual registration to make the MBS and Hybrid Web Container 2.0 and 2.0 ESD #1 clients connect to the 2.1 ESD #3 server.

CR 686034	**Incorrect error message appears when the Sybase Unwired Platform 2.1 installer does not match the system bit version.**
	If you attempt to run the 32-bit installer on a 64-bit system, you see:
	`Your system requires the 32-bit installer of Sybase Unwired Platform.`
	If you attempt to run the 64-bit installer on a 32-bit system, you see:
	`Your system requires the 64-bit installer of Sybase Unwired Platform.`
	Workaround: None. Use the installer that matches your system bit version.

CR 679421	**Communication port number conflicts may occur when installing Sybase Unwired Platform 2.1 if you indicate consecutive port numbers for the Server Administration, Data Change Notification, and Replication ports.** The Sybase Unwired Platform Runtime installer automatically creates secure port numbers by incrementing the unsecured port numbers (indicated during installation) by one. **Workaround**: To avoid port number conflicts, do not use consecutive port numbers for the unsecured communication ports.
CR 692901	**Requirements for using SAP® JCO 3 on Windows XP** If you plan to use SAP JCO 3 in Unwired WorkSpace running on Windows XP, JCO 3 requires Microsoft Visual C++ 2005 Service Pack 1 Redistributable. **Note:** This is not a requirement for Windows 7, or 64-bit Windows Server 2008 system. **Workaround**: Install the Microsoft Visual C++ 2005 Service Pack 1 Redistributable. Obtain the Microsoft Visual C++ 2005 Service Pack 1 Redistributable from: *http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=14431*. You can install the service pack after installing the 2.1 ESD #3 upgrade if you discover you need it.

Known Issues for Security

Learn about known issues and apply workarounds for Unwired Platform Security.

Issue Number	Description
CR 708833	**External authentication token is not properly handled by iOS Hybrid Web Container (HWC)**. For an external token to be passed to and used by iOS Hybrid Web Container for performing SSO, make the call to setHttpHeaders before starting the client engine by placing `[self setHttpHeaders]` in the first line in the startEngine function. See *Setting HTTP Headers* in *Developer Guide: Mobile Workflow Packages*.
CR 708623	**Credentials screen does not reappear after entering wrong data for Workflow DCN** When using SiteMinder and checkImpersonation, a user is authenticated using both a user name and token. SiteMinder checks if the user name matches the token. If the user name and token do not match, the user cannot pass the SiteMinder authentication. **Workaround**: None. Works as designed. Will be re-evaluated in a future release.
CR 703706	**Login fails with DIGEST-MD5 authentication method in LDAPLoginModule**. **Workaround**: Use the `DigestMD5AuthenticationFormat` property and set it to `DN`. You can only use a user name for the `DigestMD5AuthenticationFormat` property if the LDAP server supports it. User name formats like u:username are not supported. When using the DN, remember to use a distinguished name instead of user name for the `BindDN` property. See *LDAP Configuration Properties* in *Security*.

Known Issues for Unwired Server

Learn about known issues and apply workarounds for Unwired Server.

Issue Number	Description

N/A	**Receive `Error: 584 Message: Server public key verification failed` when running the diagtool.exe tool.**
	
	Workaround: Resolve the issue by performing one of the following:
	
	• Run the **diagtool.exe** tool from another host.
	• Edit the registry on the machine where the **diagtool.exe**tool is being run:
	  • Remove the `HKCU\Software\Sybase\Messaging Client` registry key
	  • Remove the `HKCU\Software\Sybase\MessagingClientLib` registry key
	
	Note: Use caution when editing the registry. Ensure that the key entries removed do not contain any currently used SUP applications (sub-entries).

CR 709892	**Unwired Server records deployment errors in the domain log rather than in the Unwired Server logs.** **Workaround**: None. Deployment errors are in the domain log.
CR 708431	**Data on the EIS, cache, and device is not in sync.** If a Create operation succeeds at the enterprise information system (EIS) back end, but its corresponding Entity Read operation fails at the back end, the back-end is updated but the cache database (CDB) is not. Error information may be reported in the Unwired Server log, but the error is not propagated to the device client. **LogRecord** shows 0 (zero), but **findAll** does not show the new record on the device. **Workaround**: None. The change appears in the cache database and the device when the CDB is refreshed (for example, when the cache interval expires).
CR 707989	**UserName appends @security in Application connection in manual registration**. **Workaround**: Set user name as "supAdmin@admin" in application connection if you set SecurityConfiguration in test code. The user names must match.
CR 705862	**Device client cannot connect after it is restored**. When a device MBO package database is re-created, and a backup containing the old device information is restored, the device does not receive any notifications or messaging-based messages, and no user error is visible. **Workaround**: Reregister the device.

Issue Number	Description
CR 704854	**Receive `Certificate Chain previously rejected` when registering using HTTPS to Relay Server on Android 2.3.3 and Android 3.2**. **Workaround**: None. Works as designed. This is a platform limitation with Android 2.3.3 and Android 3.2.
CR 704253	**During initial synchronization, the server updating the domainlog DB fails with a SQL Anywhere Error -638** **Workaround**: None. There is no impact from this error and it can be ignored.
CR 703612	**Unable to properly update the CDB because the partition was improperly assigned when the row was initially inserted into the cache.** An MBO with a load/synchronization parameter, using an on-demand cache group interval greater than 0, and the load parameter does not have the "Propagate to Attribute" set: **1.** A record is created by the device and synchronized to the EIS. **2.** The record is deleted in the EIS. **3.** When the on-demand cache group interval expires, the next synchronization request from the device should correctly refresh the CDB and device. Although the newly created record has been deleted from the EIS, it still exists in the CDB and device where they remain. **Workaround**:When designing the MBO, map its load parameter to a corresponding attribute with "Propagate to Attribute". If there is no corresponding attribute, add one, then map it. For example, the Customer MBO has this definition: `SELECT id, fname, lname, address, city, state, zip, phone, company_name FROM sampledb.dba.customer WHERE state = :state` Map the Customer MBO state load argument to the state "Propagate to Attribute" from the Load Arguments tab.

Issue Number	Description
CR 703515	**LDAP role/authentication search base cannot support special characters**. The following characters have special meaning when they appear in a name in LDAP: , (comma), = (equals), + (plus), < (less than), > (greater than), # (number sign), ; (semi-colon), \ (backslash), and " (quotation mark) LDAP providers do not currently handle these special characters in any of the names or DNs, including username, DefaultSearchBase, AuthenticationSearchBase, RoleSearch-Base, AuthenticationFilter, and RoleFilter. **Workaround**: None.
CR 701975	**When deploying a package from Unwired WorkSpace or Sybase Control Center using UPDATE mode, the package's "onDemand" coherence window is not up-dated on Unwired Server.** **Workaround**: After deployment, adjust the "onDemand" coherence window from Sybase Control Center.
CR 699590	**Large messages processed incorrectly on 32-bit runtime installations.** A 32-bit Unwired Server is not supported when developing messaging applications that process large messages. 32-bit servers run into memory issues and can potentially further result in incorrect data synchronizing with the Unwired Server cache. **Workaround:** Install Unwired Platform runtime components on 64-bit hosts.

Issue Number	Description
CR 699343	**For native Object API applications, the upgrade process silently deletes composite orphans from the server cache.** Unwired Platform no longer allows composite orphans – MBO child entities within a composite relationship that have no parent entity. Depending on the system configuration, the upgrade may physically delete the orphans, or may only logically delete them, then physically delete them during a subsequent cache purge process. Either way, applications that use MBOs with composite orphans may not behave the same after migration. **Workaround**: Before upgrading, eliminate composite orphans in native Object API applications: • Look for existing MBOs that generate composite orphans. Examine each MBO in Unwired WorkSpace, looking for any instance in which the application loads child entities before the related parent entity. • For any MBOs you find that generate composite orphans, change the data model in the MBO so that the application always loads parent entities before the related child entities. See the "Composite relationship behavior" row in the *Relationship Guidelines and Restrictions* topic, in the *Sybase Unwired WorkSpace - Mobile Business Object Development* guide. • Test any changes you make to ensure that the application still behaves as expected. • After you are satisfied that you have eliminated composite orphans from your MBOs, proceed with the upgrade to Unwired Platform version 2.1 ESD #3. **Additional Information on Composite Orphans** An example of a scenario in which parentless child entities can occur is a two MBO parent-child model integrated with JDBC for sales orders and line items. If the data model specifies that line items are inserted first, the application attempts to load child line items for a sales order that does not yet exist. Those line items disappear from the cache before the application can load the parent sales order. To be sure that composite orphans do not cause problems in your migrated applications: • After migration, if you have mobile application projects that contain composite relationships and the child MBO load operation has no parameter dependencies on its parent, redeploy the package from Unwired WorkSpace using Update mode to generate the correct loadGroups for the deployed package. • Child load operations that do not depend upon parameters from the parent do not always generate orphans. Consider the case where the user wants to load all sales orders and line items created in the New York office. New York may be a load parameter for both parent and child that is bound to a personalization key and results in no orphans even though the child does not get any load parameters from the parent.

Issue Number	Description
CR 698266	**Synchronization fails when QUEUED_MESSAGES.adm file gets too large**. **Workaround**: Regularly run **packtool.exe** to defragment and reclaim unused space within tables and memo files used by the Advantage Database server.
CR 695671	**Listview object query fails with error.** When performing a listview object query with very large data sets, the client returns an error code 571 or 1053, and the server becomes temporarily unavailable, or tries to restart. The `mlsrv_err.log` reports `java.lang.OutOfMemoryError`. **Workaround**: Modify the maximum Java heap size to handle very large amounts of data in an object query. See *Configuring Unwired Server Performance Properties* in *System Administration*.
CR 693116 and CR 692218	**Devices on 3G/4G network not able to connect to Unwired Server.** A connectivity issue may occur between the device and Unwired Server. The current client libraries use a protocol that supports HTTP chunked transfer-encoding only. 3G/4G traffic might, in some cases, be subject to content filtering and convert the messages from chunked transfer-encoding to fixed content-length if the communication is done via standard HTTP ports, which most commonly are 80 and 8080. Since the client application can support only chunked transfer-encoding, the communication cannot be established. In effect, a response cannot be sent back to the server indicating that the device is online. **Note:** For devices on WiFi networks, the connection between a device and Unwired Server establishes without error. **Workaround**: Use a nonstandard HTTP port (any port other than 80 or 8080) or an HTTPS port, that is, 443 for Unwired Server client communication from devices to the Relay Server or external load balancer, whichever one is used, in the DMZ.
CR 692374	**Enabling compression incurs additional processing on the device and the server side to optimize the amount of data to be transferred over the network**. **Workaround**: Carefully evaluate whether the compression trade-off is appropriate in the deployment environment, and carefully size the server capacity to support it. If appropriate in a particular environment, follow the Sybase Unwired Platform *Developer Guides* to enable compression.
CR 691923	**Existing client code not retrieving records.** Existing client code does not retrieve any records after you add sync parameters, and re-deploy the client. **Workaround**: Update the MBO, and regenerate the code. Add the `syncparamClass.save()` method in the client code.

Issue Number	Description
CR 691517	Adding new operations onto a mobile business object in a deployed package using the **UPDATE** deployment method may render client applications based on the previous package model version, nonfunctional. **Workaround**: None.
CR 686043	**In a disabled domain, messaging clients continue to receive messages from server.** **Workaround**: To disable any outbound activity to a client, lock the application connection associated with that client.
CR 576726	**The full range of valid values is not supported if an attribute uses the default datatype (`int`) for a SQL Anywhere® database column of type `unsigned int`.** **Workaround**: To support the full range of valid values for an `unsigned int` attribute in the EIS, set the datatype for the corresponding MBO attribute as `long`.

Known Issues for Sybase Control Center

Learn about known issues and apply workarounds for Sybase Control Center.

Issue Number	Description
n/a	**Domains and Packages tab** Sybase Control Center for Online Data Proxy does not have a way to add a new domain to an application from the **Application Properties** dialog. Domains can only be associated in the **Application Creation** wizard.
CR 708155	**"DOECNoAuth" security configuration assigned automatically when deploying for the first time in the domain** When deploying for the first time in the domain, the "DOECNoAuth" security configuration is assigned, even if you selected another value. **Workaround**: After deployment, change the security configuration in Sybase Control Center. 1. Navigate to **Domains** > ***DomainName*** > **Packages** > ***PackageName*** > **Settings** > **Security Configuration** 2. Select the correct security configuration, and save the changes.
CR 707966	**When deleting a domain or undeploying a DOE-C package, the server log shows a `Failed to undeploy package` error.** **Workaround**: Ignore the error.

Issue Number	Description
CR 707565	**Domain log purging can take a significant amount of time if too much data accumulates.** **Workaround**: Purge the domain log at regular intervals.
CR 705003	**SCC does not ask user to delete packages and connections before deleting the domain.** **Workaround**: None. Make sure package and connection deletion is not an issue before deleting the domain.
CR 704917	**If an administrator takes action from Sybase Control Center while the primary server is being shut down, you may see an internal or unexpected error if the server has not completely shut down.** **Workaround**: Ensure that the primary server shutdown is complete and another server has taken over as the new primary before taking any actions in Sybase Control Center.
CR 704913	**When an iOS device goes offline, the device is still shown as being online for about 8 minutes in Sybase Control Center.** **Workaround**: None.
CR 704758	**Cannot delete replication devices in Sybase Control Center; license information incorrect.** **Workaround:** Patch 01 fixes this issue. See the *Fixed Issues in Patch 01* in the *Sybase Unwired Platform 2.1 ESD #3 Patch 01* section of this release bulletin. If you cannot apply the patch, use this workaround. Upgrade the client runtime to the most current version of Unwired Platform. Starting with version 2.1, device tracking changed to application connections tracking. However, replication applications must use the 2.1.2 client runtime (or 2.1.1 for Android applications) to be compatible with current versions of Sybase Control Center and license tracking. With these clients, replication devices are "implicitly" tracked at the package level and the user is associated with the device is tracked at the package user level . The expectation is that when a package user is deleted, and if that package user no longer uses any other package from that device, the device is deleted. The license count correctly decrements by 1. A similar mechanism is in place for subscriptions.

Issue Number	Description
CR 703367	**Domain log messages in Data Sync and Operation Replay sub-systems are not logged.** **Workaround:** Use this: • Applies to the 'default' domain only. • Create a domain log profile for the 'default' server connection (JDBC type) to enable logging of Data Sync and Operation Replay messages. • If you have a similar configuration, make changes as appropriate. Please contact technical support for guidance.
CR 703321	**Cannot export domain log after upgrading.** Exporting a domain log after upgrading to version 2.1 ESD #3 generates a `Domain log export failed` error message. **Workaround**: Clear the cache for the browser you are using to access Sybase Control Center.
CR 696373	**When retrieving a domain log for large data, you receive an unknown error in Sybase Control Center.** **Workaround**: Check whether the Unwired Server or Sybase Control Center `agent.log` reports an out of memory error. If so, increase the JVM heap size appropriately for Unwired Server. Use either the Sybase Control Center Server Configuration tab or the Sybase Control Center `...\SCC_X-X\bin\ scc.properties` file). If this does not resolve the issue, try decreasing the time range set in the domain log in Sybase Control Center to a smaller value.
CR 695792	**Unwired Server stops responding when suspending DOE-C package subscriptions.** Suspending a large volume (100 or more) of DOE-C package subscriptions may cause Unwired Server to stop responding. **Workaround**: Suspend subscriptions in batches of no more than 50.
CR 691963	**MBO package deployment in Update mode after adding a new load argument without default value succeeds even though deployment should fail.** **Workaround**: None. To avoid deploying client-incompatible changes, Sybase recommends using Verify mode deployment; if verify succeeds, proceed to Update mode deployment.

Issue Number	Description
CR 690482	**Using Firefox, the Sybase Control Center main page shows `Invalid Login` and the agent.log shows `Received fatal alert: bad_certificate` error.** **Workaround**: The Firefox browser appears to be using an incorrect server certificate. Clean up any such previously existing certificates by selecting **Tools > Options > Advanced > View Certificates**. Click the **Servers** tab, then identify existing certificates issued to the same host to which you are connecting . Click **Delete** to remove all such certificates. Alternatively, use Internet Explorer.
CR 683775	**In some instances, when using the Windows Services tool or the Stop Sybase Unwired Platform Services shortcut to stop the Sybase Control Center 3.2 service, Sybase Control Center displays `Stopping` in the Windows Control Panel yet fails to stop.** **Workaround:** To stop the Sybase Control Center service: 1. Launch Windows Task Manager. 2. Locate the `sccservice.exe *32` process, right-click and select **End Process Tree**. This option also kills the Sybase Control Center database repository server process `dbsrv11.exe *32`. For 32-bit machines, `*32` does not appear.

Known Issues for Sybase SAP DOE Connector

Learn about known issues and apply workarounds for Sybase SAP® DOE Connector (DOE-C).

Issue Number	Description
CR 709691	**SUP server crashes during subscription SUSPEND/RESUME operations.** **Workaround**: Go to **SCC > Servers >** ***primary node*** **> Server configuration > general > performance configuration** and set the default value for **Maximum number of in memory messages** to 100 to avoid the OutOfMemory/Server crash issue.
CR 708754	**When load balancer is shut down, due to HTTP 404 error, DOE-C package subscriptions may be automatically removed.** **Workaround:** 1. Implement SAP note number 1722501 to prevent DOE from automatically remove the subscribed users. 2. Create a scheduled job to execute the report, SDOE_RESTART_OUTBOUND_QUEUES to restart the queues in regular intervals in the DOE system.

Issue Number	Description
CR 706532	**Suspend/Resume of DOE-C subscriptions can result in `HTTP 502 - Bad gateway` error in DOE when Load Balancer is between SUP and DOE.** **Workaround:** RESUME just a few subscriptions at one time. For example, SUSPEND 100 users, but only RESUME 10 subscriptions in one operation, then repeat that procedure 10 times.
CR 706111	**DOE-C error: `HTTP 400, Unable To Parse Request` while Resuming more than 100 users at a time.** **Workaround**: RESUME just a few users at one time. For example, SUSPEND 100 users, but only RESUME 10 users in one operation, then repeat that procedure 10 times.
CR 705888	**Domain deletion may cause HTTP 500 error in DOE-C device queue.** When DOE sends data to a subscription for a device for which the domain has already been deleted from Unwired Platform, Unwired Platform should send a recoverable HTTP 404 error, indicating that either the subscription or the domain were not found. Instead the DOE device queue shows an unrecoverable internal server error (HTTP 500) and the Unwired Server log file has a `java.lang.IllegalStateException: domain already cleared` error. **Workaround**: Remove the subscription in DOE.
CR 703909	**When DOE-C sets domain log context properties, it leaves the "domain name" field blank in the server log entry.** **Workaround**: The domain name is set correctly in the DomainLog table.
CR 700957	**If the DOE server is down, the device gets a long HTML exception.** **Workaround**: None.

Issue Number	Description

CR 626975	**If you uninstall DOE-C, Unwired Server does not start.**
	Workaround: After uninstalling DOE-C from the Unwired Server machine and re-starting that machine, the Unwired Server may start with errors or DOECAdminExceptions if there are non-DOE-C packages deployed to the Unwired Server. These errors are reported in the Unwired Server log file.
	To remove the `DOECAdminExceptions`:
	1. In *<UnwiredPlatform_InstallDir>*`UnwiredPlatform\Servers\UnwiredServer\Repository\Instance\com\sybase\djc\server\ApplicationServer\default.properties`, remove the `ResyncDOESubscriptions` property.
	2. Run `configure-mms %COMPUTERNAME%`.
	Note: If Unwired Platform has been installed into a simple Unwired Platform cluster, without failover, the `configure-mms.bat` command is as follows:
	`configure-mms <ClusterName> custom <Data Tier Hostname> <CDB port#> <CDB DB user> <CDB User password> <CDB DB Server Name> <Cluster DB Server Name> <SUP Server Admin password> <CacheDB password> <ClusterDB password> <MonitorDB password> <DomainLogDB password> <SUP Server Node running this command>`
	For example:
	`configure-mms myCoCluster custom cdbhost 5200 dba sql cdbhost_primary clusterdb_myCoCluster s3pAdmin sql sql sql supnode2`
	3. After configuration has completed, stop the Unwired Server services, then restart the Unwired Server services.
	Unwired Server finishes the configuration process during Unwired Server start-up.

Documentation Issues and Updates for Unwired Platform Runtime 2.1 ESD #3

Read about updates, corrections, and clarifications to the documentation released with Sybase Unwired Platform Runtime.

Issue Number	Description
SMPONP-5570	**Increase -nc value for ML if connections hang** If you see connections hanging and an HTTP 400 error during testing with a relay server, increase the value for the -nc mlsrv12 option to a higher number, for example, 10000. The -nc option for ML specifies the maximum number of concurrent network connections to ML. See the *MobiLink - Server Administration* guide for more information.
SMPONP-6923	**Changes to procedure for making multiple hybrid apps co-exist** See *Using Multiple Hybrid Web Containers on the Same iOS Device* on page 106.
SMPONP-5946	**Security x.509 does not work correctly** The security module is configured with the technical user and password. When this is done in the SUP log instead of SUP server using the client user ID and certificate to authenticate, it uses the technical user. **Workaround:** Keep the login and password fields blank in the connection pool to the back-end SAP.
SMPONP-5638	**New topic for troubleshooting APNS notification errors** See *Troubleshoot APNS Notification Errors* on page 108 to learn how to troubleshoot errors that are logged in the Messaging Server APNSProvider module log.
SMPONP-5646	**SUPDT is not a full qualified host name** The full qualified host name or network name should be `SUPDT.example.com` and the example should be `hostname.example.com`. Applies to *Selecting Installation Options* in the *Installation Guide for Runtime*
CR 738781	**Current versions of Afaria do not support any management via Sybase Control Center** The topic *Launching Afaria from Sybase Control Center* in *System Administration* is invalid and should be removed. The current versions of Afaria do not support management via Sybase Control Center.

<table>
<tr><th>Issue Number</th><th>Description</th></tr>
<tr><td>CR 736342</td><td>Documentation specifies the wrong directory for database driver files

In Sybase Unwired WorkSpace - Mobile Business Object Development, the documentation specifies the wrong directories for the JDBC drivers.

Workaround: The drivers should be saved to these locations:

<table>
<tr><th>JDBC driver for:</th><th>Action</th></tr>
<tr><td>Oracle</td><td>Place the JDBC driver, for example <code>ojdbc14.jar</code>, in:

<code>SUP_HOME\MobileSDK<Version>
\MobileWorkSpace\mobile
\eclipse\plugins\com.syb-
ase.uep.com.oracle_<ver-
sion>.<plugin version number>
\lib</code></td></tr>
<tr><td>DB2</td><td>Unzip the <code>db2JdbcJars.zip</code> file and copy the JAR files to:

<code>SUP_HOME\MobileSDK<Version>
\MobileWorkSpace\mobile
\eclipse\plugins\com.syb-
ase.uep.com.db2_<ver-
sion>.<plugin version number>
\lib</code></td></tr>
<tr><td>SQL Server JDBC driver 3.0</td><td>Copy <code>sqljdbc4.jar</code> to:

<code>SUP_HOME\MobileSDK<Version>
\MobileWorkSpace\mobile
\eclipse\plugins\com.syb-
ase.uep.com.sqlserver_<ver-
sion>.<plugin version number>
\lib</code></td></tr>
</table>
</td></tr>
<tr><td>CR 735927</td><td>Best practice for exporting domains

An internal integer ID is auto-generated and assigned to a new domain when it is created and that ID is not recycled/re-used when the domain is deleted.

Applies to Sybase Control Center for Sybase Unwired Platform 2.1 ESD#3 > Deploy > MBO Packages > Deploying MBO Packages > Deploying with Export and Import > Import Export Best Practices.</td></tr>
</table>

Issue Number	Description
n/a	**URL suffix property value** In the *Installation Guide for Runtime* topic *Creating a Quick Configuration* the value you would use for IIS is `/ias_relay_server/server/rs_server.dll`.
n/a	**Proxy settings** In the *Installation Guide for Runtime* topic *Creating a Quick Configuration*: • The proxy settings apply to all tiers, not only Online Data Proxy deployments. • If you have a proxy configured for connections from the internal networks (where the RSOE runs) out into the DMZ or internet (where RelayServer runs) then the four properties allow the RSOE to go through that proxy to connect successfully.
n/a	**Updates to the Troubleshooting guide were made.** The topic *Messaging Installer Completed with Errors* in *Troubleshooting > Troubleshoot the System > Troubleshoot the Postinstallation Configuration* was updated to eliminate the step directing users to run the **tracemigrator -i** command. Tracemigrator is a tool for importing a TraceConfig.xml configuration document into the database. It is not relevant for troubleshooting database upgrade errors.
n/a	**Protocol setting for iOS added to topic.** The protocol setting is available for iOS. You can set the protocol to either HTTP or HTTPS. See *Provisioning Settings*.
n/a	**Clarification on menu options**. Some menu options mentioned in the Sybase Control Center online help, *System Administration*, and *Security* appear only if functionality is installed or enabled. For example, the Advanced Search and Security Configuration tabs are visible only if clusters are installed. The documentation does not state this explicitly for each topic. If documentation refers to a tab that is not visible, the feature is not installed or enabled.
n/a	**Defining Certificates for SSL Encryption** In *Security > Server Security > Encrypting Other Listeners for Unwired Server Defining Certificates for SSL Encryption*, the note at the end of the task needs further information

Issue Number	Description
CR 717541	**Firewall/IDS systems require configuration** The *Troubleshooting* guide should include the following statement: If you are running a firewall/IDS system, configure it to allow SUP components to communicate on their configured ports.
CR 717016	**Install Relay Server on IIS** QuickSetup relay server installation scripts are provided in *Interactively Installing Relay Server on IIS with Scripts*.
CR 710989	**Topic contained incorrect information on application registration for 2.1.3**: *Registering Applications, Devices, and Users* in the *Security* documentation was not correctly updated to reflect current behavior for this release. Review the *Registering Applications, Devices, and Users* topic in the Release Bulletin for current registration information.
CR 709894	**New connection template and usage restriction** A new application connection template named `$diagtool` is automatically added by the installer in the available templates list in Sybase Control Center. However, this template was not documented in Sybase Control Center online help. The template is a required template exclusively used for the **diagtool.bat** diagnostics utility. It is read only and cannot be deleted. Do not use it for any Unwired Platform device applications.
CR 707294	**Unwired Server port inaccessible from a remote machine.** A port in the Sybase Unwired Platform environment is inaccessible from a remote machine, which may indicate the machine is configured with more than one Ethernet adapter and IP address. Consequently, a service port on Unwired Server may be configured with an IP address that is inaccessible from the remote machine. See *Unwired Server Port Is Not Accessible from a Remote Machine* on page 110.
CR 706677	**Enabling and Configuring a Password Policy for Data Vault Logins** Administrators can create a password policy for device application logins in a new or existing application connection template.
CR 705731	**Mutual Authentication for Messaging and Unwired Server Communications on IIOPS Ports** You can no longer configure mutual authentication for messaging and Unwired Server communication over IIOPS, because Sybase Control Center and Public Administration APIs have been modified to prevent administrators from configuring the IIOPS management port with a profile that uses `mutual_auth`.

Issue Number	Description
CR 704382	**Administration Workbook contains topics that use OpenDS as an example of an LDAP directory.** Ignore references to OpenDS as Unwired Platform no longer includes this LDAP directory. Instead of using OpenDS to perform workbook activities, Sybase recommends using Microsoft Active Directory.
CR 698525 and CR 698515	**Running the Pack Tool Utility.** As part of a regular administrative procedure, run the **packtool.exe** utility to keep the system in the proper running state and to compress the ADS data file (`QUEUED_MESSAGES.adm`) for increased messaging client performance. **Important:** In a clustered environment, you must use the latest version of the **packtool.exe** utility. Contact Technical Support, or use the utility shipped with Sybase Unwired Platform 2.1 ESD #2 or later.
CR 697403	**Resetting the supAdmin Password.** Manually reset the supAdmin password if you forget it. Contact your IT department or administrator before changing or resetting the password. Only the person who has the correct permission to change the configuration files should perform the procedure.
CR 696314	**When upgrading, an unhandled exception `('System.MissingMethodException') occurred in JMSBridge.exe [8320]` is received**. This occurs only if an EAServer installation and a Sybase Unwired Platform installation coexist on the same machine. The exception is caused by different versions of the `com.sybase.iiop.net.dll` file on the system. Ensure that your environment path includes only the Sybase Unwired Platform `com.sybase.iiop.net.dll` assembly. For example, if you have EAServer installed on the same machine as Sybase Unwired Platform: 1. Use a strings command (using UNIX tools for Windows) to check if SetTrustAllCertificates exists on the assembly file that `jmsbridge.exe` loaded. From a command prompt, enter: `>strings -a com.sybase.iiop.net.dll | grep SetTrustAllCertificates` The command reports back with `SetTrustAllCertificates` if it exists within the assembly. 2. Remove the EAServer installed `com.sybase.iiop.net.dll` from the global assembly cache (GAC).

Issue Number	Description
CR 697230	**Client application cannot register to Unwired Server through Relay Server**. This troubleshooting topic applies to the 2.1 *Installation Guide for Runtime*. Problem: The Sybase Unwired Platform client application cannot register to the Unwired Server through the Relay Server, and the client displays an error such as `HTTP_TPSendRequest - returning error code -106 -- HTTP error -- header is: HTTP/1.1 404 Not Found Content-Type: text/html`. Explanation: The Relay Server cannot find the language resource file (`dblgen12.dll`). Make sure the language resource is within the system search path and that the Web server worker has permission to read it. Update the system PATH variable on the IIS host to include the `ias_relay_server \Server\` directory.

CR 697226	**Configuring IIS 7.5 for Relay Server**
	The 2.1 *Sybase Unwired Platform Release Bulletin for Runtime* explains how to configure IIS 7 for Relay Server. If you are configuring IIS 7.5, extra settings are needed.
	Follow the steps in *Configuring IIS 7 for Relay Server* in *Installation Guide for Runtime* (the *Deploying an Enterprise System* section). In step 6 (Add Relay Server locations to the default Web site), add the following information:
	`<location path="Default Web Site/rs">`
	`<system.webServer>`
	`<security>`
	`<authentication>`
	`<anonymousAuthentication userName="" />`
	`</authentication>`
	`<requestFiltering>`
	`<requestLimits maxAllowedContentLength="2147483647" />`
	`</requestFiltering>`
	`</security>`
	`<handlers accessPolicy="Execute, Script" />`
	`</system.webServer>`
	`</location>`

Using Multiple Hybrid Web Containers on the Same iOS Device

You can configure two or more Hybrid Web Containers to coexist on the same device.

This customization allows two or more independent users to use the same device, but with their own private version of the application. In summary, you need to change the application ID, the bundle identifier, and possibly the URL scheme.

The application ID is used by the server to identify the application, and because of this, you cannot run two applications on the same device with the same application ID. By default, the Hybrid Web Container uses “HWC” for its application ID. Changing the application ID involves a minor change to `CustomizationHelper.m`. Additionally, you must signify to iOS that this is a distinct application. This requires a minor change to update the application bundle ID in the `plist` file. Finally, if your application needs to communicate with the Afaria client for provisioning your application or retrieving a certificate, you need to specify a unique URL scheme in the `plist` file. If your application does not need to communicate with the afaria client, then you should delete the “URL types” item from the same `plist` file.

1. Change the project name:
   a) In the Xcode Project Navigator, click on the root Hybrid Web Container element.
   b) With the Hybrid Web Container element highlighted click on the Hybrid Web Container text to rename.
   c) Change the name of the Hybrid Web Container element to your new project name.
   d) A window to rename project content items appears. Click **Rename**.
2. Change the application ID:
   a) In Xcode Project Navigator, find and open the `CustomizationHelper.m` file, which is located in the `Classes` group folder,
   b) Locate the customization point that accompanies the `getAppId` function, and change it so that it returns a unique name.
   c) Save and close the file.
3. To differentiate this version of the Hybrid Web Container from another:
   a) In Xcode Project Navigator, find and open the `HWC-Info.plist` file, which is located in the `Resources` group folder.
   b) Change the bundle identifier value to something unique.
   c) Save and close the file.

   The container template project has a URL schema setting in the project `plist` file, which is used to communicate with Afaria client.
4. To avoid multiple container applications from interfering with each other when communicating with the Afaria client, the URL schema must be unique among all container applications that are installed on the device, otherwise, the application may be launched by the afaria client by mistake, or fail to launch altogether.
   a) In Xcode Project Navigator, find and open the `HWC-Info.plist` file, which is located in the `Resources` group folder.
   b) Expand the **URL types item** > **Item 0** > **URL Schemes item**.
   c) Select **Item 0**, and change its value to a unique value among all other applications.
   d) Save and close the file.

Troubleshoot APNS Notification Errors

Resolve runtime errors reported in the APNS Notification log file located in *SUP_HOME*\Servers\UnwiredServer\logs\APNSProvider, by taking the recommended action.

- **20130502T131941.847|1| - Feedback Error: A call to SSPI failed, see inner exception. 20130502T131947.647|1|Error. FeedBack ensureconnected exception. A call to SSPI failed, see inner exception –** recommendation: from Sybase Control Center, confirm that the certificate used in Apple push configuration has not expired.
- **20130516T175645.847|1|Device Token length is not 64 –** recommendation: an empty token in the application connection due to a timing related error that should correct itself once the token arrives from the device. It occurs when a notification is sent to the device in the time-frame between the application connection getting activated when the token is empty and APNS token getting sent to the server from device.
- **Error: Unable to write data to the transport connection: An existing connection was forcibly closed by the remote host –** recommendation: this error is seen when the token length is empty or an expired certificate is used. This error may also be logged in cases of occasional issues with the Apple push service itself.

Messaging Installer Completed With Errors

When upgrading to a newer software level, you are informed of errors.

```
"***Error!!! Error!!!***
The Messaging installer completed with the following errors. These
errors will have to
be addressed before the services become functional.
Please refer to documentation for recovery process.
Database upgrade failed."
```

Explanation: This error happens when the Messaging server database upgrade does not complete. The most likely reason for this is that the Messaging server database had some lingering connections to a table while the upgrade was running. You can get more detailed information from the trace file located in *UnwiredPlatform_InstallDir*\UnwiredPlatform\Servers\UnwiredServer\Logs\DBUpgrader.

1. Stop all Sybase Unwired server services in all the application nodes.
2. Restart the database services in the data tier node. This is to make sure the Messaging Database has no open connections.
3. From the primary server node, open a Windows command window and navigate to *UnwiredPlatform_InstallDir*\UnwiredPlatform\Servers\MessagingServer\Bin.
4. Run the command **dbupgrader –v current** and make sure no error messages are displayed.
5. Start the Sybase Unwired Server service in the application node and verify it can start.

Interactively Installing Relay Server on IIS with Scripts

(Recommended) Use quick setup scripts to interactively install Relay Server. Quick setup can be less error-prone than manual installations.

Prerequisites

Follow the prerequisites identified in the quick setup script. You cannot install Relay Server until the script verifies that the prerequisites have been met.

Task

Output of this setup script is saved to `rs-setup.log`. The existing `rs-setup.log`, IIS metabase, and Relay Server configuration files are backed up automatically.

1. Locate the quick setup script for your version of IIS, and review the `readme.txt` file for your IIS version.
   - Launch `rs-setup.bat` for IIS 6 from *`SUP_HOME`*`\Servers\SQLAnywhere16\MobiLink\relayserver\IIS\QuickSetup_IIS6`. For information about this script, see *http://dcx.sybase.com/index.html#sa160/en/relayserver/ml-relayserver-s-5692444.html*.
   - Launch `rs-setup.bat` for IIS 7 from *`SUP_HOME`*`\Servers\SQLAnywhere16\MobiLink\relayserver\IIS\QuickSetup_IIS7`. For information about this script, see *http://dcx.sybase.com/index.html#sa160/en/relayserver/ml-relayserver-s-5692444a.html*
2. Follow the prompts to install files in the correct location and to configure IIS for Relay Server use.

   The script guides you through:
   - IIS customization
   - Backup creation
   - Installation and Relay Server startup
   - Generation and launch of a Quick Reference document
   - Generation and launch of a status page
   - Launch of a SimpleTestApp client

Registering Applications, Devices, and Users

Before any application can access the runtime, the user, device, and application must be identified by first registering with Unwired Server and pairing them with a device and user entry. Only when all three entities are known, can subscriptions can be made or data synchronized.

In Sybase Control Center, Platform administrators set up an application connection template for applications . Part of this template includes a property that enables automatic registration.

- When automatic registration is enabled, a device user need only provide valid Unwired Platform credentials that are defined as part of the security configuration.
- When automatic registration is disabled, the platform administrator must provide the user a username/passcode out-of-band. This is the passcode initially required by login screens to access the application for the first time, and expires within a predetermined time period (72 hours, by default).

Note: Choose to use automatic registrations carefully, especially if there are multiple application connection templates for the same application. The combined criteria of the application ID and security configuration used by the device application trigger a search for a matching template that completes the automatic registration. However, if the security configuration is not sent by the device application, and server finds multiple templates, registration fails.

Unwired Server Port is Not Accessible from a Remote Machine

Problem: A port in the Sybase Unwired Platform environment is not accessible from a remote machine.

Explanation: This may indicate the machine is configured with more than one Ethernet adapters/IP addresses. Consequently a service port on Unwired Server may be configured with an IP address that is not accessible from the remote machine.

Solution:

1. From the command line, issue **ipconfig** to find out more information.

```
Windows IP Configuration
Ethernet adapter Local Area Connection* 11:
   Media State . . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
Ethernet adapter Local Area Connection 2:
   Connection-specific DNS Suffix  . : sybase.com
   Link-local IPv6 Address . . . . . : xxxx:xxxx:xxxx:xxxx
   IPv4 Address. . . . . . . . . . . : xxx.xxx.xxx.xxx
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : xxx.xxx.xxx.xxx
Ethernet adapter Local Area Connection:
   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : xxxx:xxxx:xxxx:xxxx
   Autoconfiguration IPv4 Address. . : xxx.xxx.xxx.xxx
   Subnet Mask . . . . . . . . . . . : 255.255.0.0
   Default Gateway . . . . . . . . . :
Tunnel adapter Local Area Connection* 8:
   Media State . . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . : sybase.com
Tunnel adapter Local Area Connection* 9:
```

2. Check for multiple Ethernet adapters and IP addresses assigned to the machine.
3. Disable any unidentified Ethernet networks.
4. Retry connecting to the port from the remote machine.

Running the Pack Tool Utility

As part of a regular administrative procedure, run the Pack Tool utility to defragment and reclaim unused space within tables and memo files that the Advantage Database Server® (ADS) data file uses. This will ensure that Unwired Server continues to run in the proper state.

Prerequisites

Ensure that you have sufficient free disk space for the packing process. The free disk space must be at least the size of the largest table.

- Single node installations – check in *UnwiredPlatform_InstallDir*\UnwiredPlatform\Servers\MessagingServer\Data\OBR
- Cluster installations – check in *UnwiredPlatform_InstallDir*\UnwiredPlatform\Data\Messaging\OBR

You need to look for the largest table, including the sum of all three table files with the extensions `.adt`, `.adi`, and `.adm`. Not all tables have corresponding `.adi` and `.adm` files.

Task

1. Stop all Sybase Unwired Platform services including the Advantage Database Server service.

   In a clustered installation, make sure services on all the nodes are stopped.
2. Back up your ADS database files (all the files in the `OBR` folder).
3. Start the Advantage Database Server service only.

   All Sybase Unwired Platform services except Advantage Database Server service must remain stopped until the pack is complete.
4. Open a command prompt window.

   If this is a cluster installation, do this in any of the application server nodes.

   If you are running Vista, Windows 7, or Windows 2008, make sure you run the command prompt as the Administrator.
5. Navigate to *UnwiredPlatform_InstallDir*\UnwiredPlatform\Servers\MessagingServer\Bin.
6. Run **packtool.exe**, with no arguments.

After running **packtool.exe**, a new log file is created in *UnwiredPlatform_InstallDir*\UnwiredPlatform\Servers\MessagingServer\Bin in the format `PackTool_`*yyyyMMddTHHmmss*`.log`.

Resetting the supAdmin Password

You can manually reset the current platform administration password.

This procedure is for Sybase Unwired Platform version 2.x.x or later.

Note: You must contact your IT department or administrator before changing or resetting the password. Only the person who has the right permission to change these files should perform this procedure.

1. Open the Unwired Server `default.xml` file, located in `<UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\UnwiredServer\Repository\CSI\conf` and modify this line:

   ```
   <options encrypted="false" name="password"
   value="{TXT:}s3pAdmin" />
   ```

   In this example, you are setting the new password to `s3pAdmin`. You can replace this password with any password you choose. Do not remove the `{TXT:}` prefix to the password.

   Note: In this example, password encryption is set to false. Disregard this value; you will configure encryption correctly in step 6.

2. Save the file.
3. Restart Unwired Server and Sybase Control Center.
4. Log in to Sybase Control Center using supAdmin as the new password.
5. When login succeeds, Sybase Control Center opens the management view on the local Unwired Server.
6. In Sybase Control Center, expand the **Security** node:
   a) Click **admin**.
   b) Click **Authentication** and select **PreConfiguredUserLoginModule** for the supAdmin user.
   c) Click **Properties**, and:
      - Enter the new password. By resupplying the password here, the file is overwritten using the correct syntax.
      - Add a Clear Password property, and set its value to `false`. This property encrypts the password. Conversely, to store the password in clear text, set this value to true.
   d) Click **Save**.
   e) When you see the warning message, click **OK**, then click the **General** tab.
   f) Click **Apply**.

      You see another warning.
   g) Click **OK**.

Configuration files are rewritten using the values you entered. When the process completes, you see a `Successfully saved` message.

7. Login again to Sybase Control Center using the supAdmin login and the new password (in this example, s3pAdmin).
8. Go to the `...\UnwiredServer\Repository\CSI` folder and verify `default.xml` to verify that encryption is configured correctly, and that the password is no longer recorded in clear text.

```
<authenticationProvider controlFlag="optional"
name="com.sybase.security.core.PreConfiguredUserLoginModule">
<options name="username" value="supAdmin"/>
<options name="roles" value="SUP Administrator,SUP Domain
Administrator,SUP DCN User"/>
<options encrypted="true" name="password" value="1-
AAAAEgQQWd8NguXX5nswpWF1vUFPtcJhjmoiSYUzEAAiY3vWkZ+Y/33cWAoUD+EV/
D80Yo4vie/
XIyZVoBZbTT9ijxHDe7wbIBsagzS0DdAvS5lTRvRRNVp83+pTjQ3mmMNt5FmxrGvU
V5fVQ2JI1YaTPbd+Tw==">/>
```

Next

Remove the No Auth Login Module entries that you added to the `csi.properties` file, and, if used,the `roles-map.xml` file. Restart Sybase Control Center for the change to take effect.

Defining Certificates for SSL Encryption

In Security > Server Security > Encrypting Other Listeners for Unwired Server Defining Certificates for SSL Encryption, the note at the end of the task needs further information.

The note describes that a password change by **keytool** must match the password you configure in Sybase Control Center for SSL. However, the task neglects to mention that if you change the password, the password must also be applied to any alias that exists. You can do this with a **keytool** command similar to these examples:

```
keytool -keypasswd -alias sample1 -keypass changeit -new changeit2 -
keystore keystore.jks
```

```
keytool -keypasswd -alias sample2 -keypass changeit -new changeit2 -
keystore keystore.jks
```

Mutual Authentication for Messaging and Unwired Server Communications on IIOPS Ports

You can no longer configure mutual authentication for messaging and Unwired Server communication over IIOPS, because Sybase Control Center and Public Administration APIs have been modified to prevent administrators from configuring the IIOPS management port with a profile that uses `mutual_auth`.

Review this table to understand how mutual authentication for Sybase Control Center ports documentation is affected. Certain documented activities are not required or supported with this change.

Document	Description
Security	In *Changing Installed Certificates Used for Unwired Server and Sybase Control Center HTTPS Listeners* , disregard statements requesting you to move certificates to the Sybase Control Center truststore. This is no longer required.
Sybase Control Center Online Help	In *Unwired Server Management Ports*, disregard the note relating to `mutual_auth` and copying certificates.
	In *Creating an SSL Security Profile in Sybase Control Center* , be aware that if you use a profile for a management port, do not select `mutual_auth`. For existing profiles, this value is no longer available.

Enabling and Configuring a Password Policy for Data Vault Logins

Administrators can create a password policy for device application logins in a new or existing application connection template. A password policy ensures that user-defined passwords conform to corporate security requirements.

A policy cannot be enforced unless developers add enforcement code to the data vault for an application. For information about creating a data vault that enforces the platform policy, see *Creating a Data Vault that Enforces Password Policy*.

1. In the left navigation pane, click the **Applications** node.
2. In the right administration pane, click the **Application Connection Templates** tab.
3. Choose one of the following:
   - To create a new template with a password policy, click **New**.
   - To edit an existing template, select the name of the template and click **Properties**.
4. Assign the connection template to an application.

Password Policy Properties

Create a password policy for device application logins. Only passwords that meet the criteria of the policy can be used to access the sensitive artifacts secured inside a device's data vault.

You can create a password policy as part of an application connection template. Ensure your developers add enforcement code to the application's data vault.

- **Enabled –** Set this value to `True` to enable a password policy for device applications. By default, this property is set to True.
- **Default Password Allowed –** Set this value to `True` to allow default passwords. If a default password is allowed in the policy, developers can create the vault using with a default password, by specifying null for both the salt and password arguments. By default, this value is set to False

- **Expiration Days –** Sets the number of days the existing password can be used before it must be changed by the user. By default, this value is set to 0, or to never expire.
- **Has Digits | Lower | Special | Upper –** Determines what combination of characters must be used to create a password stringency requirements. The more complex the password, the more secure it is deemed to be. Set the value to `True` to enable one of these password stringency options. By default they are set to false.
- **Lock Timeout –** Determines how long a successfully unlocked data vault will remain open. When the timeout expires, the vault is locked, and the user must re-enter the vault password to resume using the application. Use this property in conjunction with the Retry Limit.
- **Minimum Length –** Sets how long the password chosen by the user must be. By default, this value is set to 8.
- **Minimum Unique Characters –** Determines how many unique characters must be used in the password. By default this property is set to 0. For example, if set that the password has a minimum length of 8 characters, and the number of unique characters is also 8, then no duplicate characters can be used. In this instance a password of `Sm00the!` would fail, because two zeros were used. However, `Smo0the!` would pass because the duplication has been removed.
- **Retry Limit –** Sets the number of times an incorrect password can be retried before the data vault is deleted. A deleted vault means that the database encryption key is lost, and all data in the application is rendered irretrievable. As a result the application becomes unusable. By default this value is set to 20.

Provisioning Settings

Enter settings in the provisioning file for connecting to Unwired Server.

The provisioning file is an ASCII text file where each setting must be in a separate line, or separated by a semicolon. All fields in the configuration file are optional—that is, whatever is present in the file is imported.

Key	Description
servername=	The server name for the machine that hosts the Unwired server, or relay server, if used, where the mobile application project is deployed.
serverport=	The server port number for Unwired Server. The default is 5001
companyid=	The company or farm ID you entered when you registered the device in Sybase Control Center, in this case, 0 (zero).

Key	Description
username=	The name of the user to which the package is assigned. This should be the user name used in the registration.
activationcode=	The activation code for the registered user, for example, `123`.
serververificationkey=	The server verification key, if present, must be a valid Base64 encoded string of the messaging server's public key. An invalid Base64 string results in an empty key.
autoreghint=	This value is used exclusively for the SUP administrator to convey information to the device application; Unwired Server runtime does not use this value. The application developer uses this setting at runtime to indicate how the application is initially registered – either automatically or manually. • 1 – Always use automatic application registration. The application user is not prompted to enter an activation code upon initial activation. • 2 – Always use manual device registration. The application user is prompted to enter an activation code upon initial activation. This implies that automatic device application registration is not enabled on Unwired Server. • 0 – Both modes are supported. The application user interface allows automatic registration , or requires an activation code for manual registration.

Key	Description
urlsuffix=	The URL suffix is a pattern that is used internally by Sybase Unwired Platform when constructing URLs as part of the correct path to connect to Unwired Server. By default, this property does not need to be set. The device client internally auto-detects the correct path when connecting directly to Unwired Server, or when connecting to Unwired Server via the Relay Server using its default settings. **Note:** You only need to explicitly configure this property to match the modified URL suffix, if the default path and URL suffix of the Relay Server is modified.
protocol=	**Note:** This applies to iOS only. The protocol used to connect to the Relay Server or the reverse proxy server. Can be set to protocol=HTTP or protocol=HTTPS. The default is HTTP.

Known Issues for Sybase Mobile SDK 2.1 ESD #3

Learn about known issues and apply workarounds for Sybase Mobile SDK.

Known Issues with Installing, Uninstalling, and Upgrading

Learn about known SDK installation, uninstallation and upgrade issues and workarounds.

See the *Sybase Unwired Platform Installation Guide for Sybase Mobile SDK* for installation and information regarding the SDK

Issue Number	Description
CR 709104	**JDK and SySAM license files** When a system has Unwired Platform Runtime 2.1, Mobile SDK 2.1, and Mobile SDK 2.1.3 installed, and Unwired Platform Runtime 2.1 is uninstalled with Mobile SDK 2.1 and 2.1.3 installations intact, the JDK installation and the SySAM license file shared with Mobile SDK 2.1 installation will be removed.

Issue Number	Description
CR 709102	**Unwired Platform 2.1 Installer** When Runtime 2.1 installer is launched on the system with Mobile SDK 2.1.3 or newer installed already, user will see “null” as the installation destination path which is not changeable. **Workaround 1**: Install 2.1 runtime before Mobile SDK. **Workaround 2**: If Mobile SDK is already installed, do the following: • **32-bit**: Prior to installing 2.1, in the Windows Registry, rename HKLM\Software\Sybase\MobileSDK\2.1.3 to HKLM\Software\Sybase\MobileSDK\2.1. Once the 2.1 runtime install is complete, undo the rename. • **64-bit**: Prior to installing 2.1, in the Windows Registry, rename HKLM\Software\Wow6432Node\Sybase\MobileSDK\2.1.3 to HKLM\Software\Wow6432Node\Sybase\MobileSDK\2.1 in the Windows Registry. Once the 2.1 runtime install is complete, undo the rename.

Known Issues for Sybase Unwired WorkSpace – Mobile Business Object Development

Learn about known issues and apply workarounds for Unwired WorkSpace and mobile business object (MBO) development.

Issue Number	Description
CR 709701	**Issue**: Mapping an MBO create, update, or delete operation parameter to both a personalization key and a client parameter may cause failure on device. If both are mapped, the client parameter value is ignored and the personalization key value is used. Unwired WorkSpace may show this warning: `Client parameter 'GENERALDATA' might not be used, as the mapped argument has 'Fill from Attribute' or 'Personalization Key' specified.` Although deployment and code generation succeeds, you may still get a `NullPointerException` in the server log during runtime and the operation fails on the device. **Workaround**: Only map create, update, or delete operation parameters to a personalization key or a client parameter, not both.
CR 709076	**Issue**: After Unwired WorkSpace crashes, the current Eclipse workspace retains inconsistent states that can cause a restart to hang. **Workaround**: This is a known Eclipse issue. Remove the following file from the problematic Eclipse workspace, then restart Unwired WorkSpace: `<The workspace used in the previous crash>\.metadata\.plugins\org.eclipse.core.resources\.snap`.

Issue Number	Description
CR 705643	**Unwired WorkSpace fails to refresh the attributes of a MBO based on a stored procedure after changing the stored procedure definition in the EIS.** **Workaround**: Re-create the JDBC MBO after changing the stored procedure definition in the EIS.
CR 705313	**Messages about missing directories and directory contents when you generate ObjectAPI code.** When the MBO developer regenerates ObjectAPI code from the Unwired WorkSpace Code Generation wizard, if **Clean up destination before code generation** is selected in the Configure Options page, after completing code generation, an error similar to this displays in the Error Log view: `Resource '/<Mobile Application Project name> /Generated Code/temp_afx does not exist. Resource ' /<Mobile Application Project name>/Generated Code/genfiles' does not exist.` **Workaround**: Ignore the error messages. The objectAPI code generates successfully.
CR 703200	**MBO operations that include old value arguments that have large object types should be ignored, but the SQL clauses still include the old value arguments** Unwired WorkSpace automatically creates an MBO with old value arguments (for operations) when you drag and drop a database table. If the datasource includes operation arguments with large object types (BigString/BigBinary), Unwired WorkSpace displays a warning message stating the old value argument with large object types are ignored, but the SQL definition generated still includes the old value argument with the large object type. **Workaround**: Manually remove any SQL clauses which include old value arguments with large object types.
CR 702696	**Adding load arguments with "Propagate to attributes" that have no corresponding datasource columns, then selecting Refresh, may fail due to missing load argument default values.** **Workaround**: Unmap the "Propagate to attribute" field, which allows you to specify the necessary default values so that the execution to the EIS successfully retrieves MBO metadata.

Issue Number	Description
CR 702691	**Calls from Unwired WorkSpace to the EIS during MBO creation/development cannot distinguish syntax from data errors.** When adding an operation and corresponding SQL statement, Unwired WorkSpace executes the SQL statement to verify it is correct. Sometimes the return error is not a syntax error but a data error (for example, a foreign key restriction). The EIS API cannot distinguish syntax from data errors and exits the wizard. **Workaround**: Be aware of the EIS schema to which you are binding the MBO. For example, verify that database data is correct, and enter the correct default values so the SQL statements execute correctly.
CR 702643	**Output mapping warning message for multilevel insert operations** During deployment, output mappings are automatically generated for Create operations to support multi-level insert operations. The set of generated output mappings may not be the same as the output mappings configured in Unwired WorkSpace for the operation. However, the final set of output mappings is a union of the two sets, so the user-configured output mappings are retained. The MBO developer may see a warning in the output mapping tab indicating that some output mappings are ignored when using the "Apply Output of ENTITY READ" operation cache policy. **Workaround**: Ignore the warning since the output mappings are automatically generated during deployment.
CR 702005	**MBO operations bound to SAP BAPIs that use the "Invalidate the cache" policy may not refresh the Unwired Server cache.** Since SAP BAPI operations run asynchronously within the SAP system, the operation may not complete before Unwired Platform attempts to refresh the cache with new or updated results if using the "Invalidate the cache" policy. **Workaround**: Do not use the "Invalidate the cache" policy with MBO operations bound to an SAP BAPI.
CR 701440	**After starting Unwired WorkSpace, Eclipse OSGI bundle related error message(s) may appear in the Error Log view. For example, org.eclipse.osgi.framework.internal.core.AbstractBundle$BundleStatusException** **Workaround**: Ignore these errors.
CR 689859	**Sybase Unwired Platform Unwired WorkSpace Welcome page moves to standby mode after Eclipse is started.** This is due to a known Eclipse 3.7 bug (*https://bugs.eclipse.org/bugs/show_bug.cgi?id=355560*). In standby mode, the Welcome page is docked to the right as a view in the Mobile Development perspective, since the Problem View is part of the Mobile Development perspective. **Workaround**: Double-click the docked Welcome page to extend it to full screen.

Issue Number	Description
CR 689707	**Default runtime values requirements.** For an MBO operation, a non-nullable argument must get its runtime value via one of the following: client parameter, personalization key, default value, or fill-from-attribute. If a non-nullable argument has fill-from-attribute or client parameter already, its default value is ignored. Unwired WorkSpace still allows the user to input a default value. **Workaround**: In this case, you need not set the default value, because it is not used during runtime.
CR 688956	**Unwired WorkSpace fails during preview or test execute** When doing preview or test execute in Unwired WorkSpace during MBO development if the data volume is large, for example, either a large number of rows, or a large object in the returned result, Unwired WorkSpace may run out of memory and fail. **Workaround**: Do not select Preview or Test Execute with large results, or increase the Java VM argument, -Xss -Xms, and -Xmx in the `UnwiredWorkSpace.bat` to increase the stack and memory (heap) size.

CR 688576	**The Oracle datatype NCLOB is not supported.**
	Workaround: Convert the unsupported datatype to a supported datatype. Convert the CLOBS datatype to a varchar datatype. There are two ways to do this:
	• Create a stored procedure and call it from the MBO. In the stored procedure, convert the CLOBS to varchar:
	• Create this Transact-SQL® statement, which fetches 4000 bytes of the CLOB: `select dbms_lob.substr( x, 4000, 1 ) from T`.
	The maximum varchar length allowed in Transact-SQL statements is 4000 bytes. You can increase this to 32KB. Sybase recommends that you use a stored procedure to process the BLOB in this way. For example:
	`declare`
	`  my_var long;`
	`begin`
	`  for x in ( select X from t )`
	`  loop`
	`  my_var := dbms_lob.substr( x.X, 32000, 1 );`
	.

CR 682651	**Unwired WorkSpace does not support saving files as PDF.** Invoking **File > Save As Image File** from the Object Diagram context menu fails if PDF is the image format. This is a known Eclipse GMF bug. See *https://bugs.eclipse.org/bugs/show_bug.cgi?id=338380* **Workaround**: None.

Issue Number	Description
CR 677453	**When Unwired WorkSpace is uninstalled and reinstalled in a different directory, you get an ClassNotFoundException message if the old workspace is used and you connect using the JDBC connection profile.** **Workaround**: Update the driver definition to point to the correct path: 1. From Unwired WorkSpace, select **Window > Preferences > Sybase, Inc > Connectivity > Driver Definitions**. 2. Select the driver type, and click **Edit**. 3. Click the **Jar List** tab, and edit the path of the JDBC driver. 4. Reimport any mobile application projects that require the updated driver for connecting to the database.
CR 676634	**For Web Service and other data sources with complex structures and deeply nested element trees, StackOverflow or OutOfMemory messages may occur, or some of the operation input parameters may not be generated properly.** In some cases, you see this error: `The assigned stack size and/or heap size is not sufficient for the attempted operation.` **Workaround**: Increase the stack and heap size of the Eclipse runtime parameter in `UnwiredWorkSpace.bat` with -Xmx, -Xms, -Xss arguments, or reduce the complexity or nested level of the Web Service definition of the MBO operation.
CR 642942	**Cannot generate code with Javadoc option enabled to a folder containing Chinese characters** When using the Code Generation wizard, if you enable the option to generate Javadoc, you cannot generate the code to a folder containing Chinese characters without first changing the language for non-Unicode programs to Chinese. **Workaround:** 1. Select **Start > Settings > Control Panel**. 2. Double-click **Regional and Language Options**, then click the **Advanced** tab. 3. From the drop-down list, select **Chinese**, and click **Apply**. 4. Click **OK**.

Known Issues for Sybase Unwired WorkSpace – Mobile Workflow Package Development

Learn about known issues and apply workarounds for Unwired WorkSpace and Mobile Workflow application development.

Issue Number	Description
CR 707123	**Uploading large image not successful on some Android devices.** This message appears after selecting an image while in a workflow, but image previews and uploads continue to work: `File type is not supported!` **Workaround**: Because this issue only occurs with the File Manager application, use a different gallery application by resetting the default: **1.** Open the Settings application. **2.** Click **Applications > Manage applications**. **3.** Click **All**. **4.** Select the current default gallery application. **5.** For the **Launch by default** setting, click **Clear defaults**.
CR 700357	**Comma added for int types when number is more than 1000.** On iOS 5.0, with the Safari mobile browser, in fields with the int type, when you enter a number larger than 1000, a comma is inserted. This is expected behavior with the Safari mobile browser. **Workaround**: Set the logical-type to text instead of numeric.
CR 700077	**BlackBerry 5 read-only slider can be modified.** The anchor can be moved on a slider that is designed to be read-only. This is a known BlackBerry problem. **Workaround**: None.
CR 699341	**BlackBerry 7 form labels do not use specified theme.** On BlackBerry 7 devices, in a form that uses labels on the left, if the first field is editable, the second label does not initially use the theme set in the form. **Workaround:** Type in the field to change the label to the appropriate theme.

Issue Number	Description
CR 698779	**DatePicker control does not reopen in BlackBerry 7+.** In a BlackBerry 7 or later device, if you invoke the DatePicker control in a workflow by clicking in a date field, click **Cancel** in the DatePicker, and then click again in the same field, the DatePicker control does not reopen. **Workaround**: After clicking **Cancel** in the DatePicker, click in another field that is not of type DATETIME or TIME. You can then click in the original date field to open the DatePicker.
CR 694665	**AsyncRequestErrorLogs key not generated in a migrated project.** If you migrated a Mobile Workflow project that previously contained a submit workflow action, setting its error screen property does not generate the AsyncRequestErrorLogs list key. **Workaround**: Create a new Submit Workflow action, or click the **Generate Error Screen** button.
CR 694652	**In Mobile Workflow applications generated with a 2.0 version of the Mobile Workflow Package generation wizard, running on an iPad OS 5 and running in a 2.1 ESD #2 or later container, edit boxes that should be editable may not be editable.** **Workaround**: Regenerate and redeploy the Mobile Workflow application using a 2.1 ESD #2 or later version of the Mobile Workflow Package generation wizard.
CR 690460	**On BlackBerry devices, in most cases, the module name is the same as the module display name, but if there is a case when the module name ! = module display name,the device cannot receive asynchronous error messages.** **Workaround**: None.

Issue Number	Description
CR 690438	**A workflow does not initiate an operation.** This could happen if the workflow is not constructed correctly. On a Listview details screen, you cannot invoke a submit workflow to invoke an MBO operation that uses child keys of the list key that is bound to that listview. To prevent developers from performing an illegal operation, the keys are removed from the list. However, you can see those keys before making the workflow type operation a Listview details screen because it is not illegal in that circumstance; it is only illegal once you make it a Listview details screen. If you use those keys when it is not a Listview details screen and then make it a Listview details screen, the workflow operation will fail with no indication of the problem. **Workaround:** **Client-initiated workflow:** verify that no online request operations (such as **findall**) precede the submit workflow type operation. If so, remove the operations. **Server-initiated workflow:** verify that no links (between the start point and the first screen saved) precede the submit workflow operation. If so, remove the links.
CR 684635	**Choice controls with a large number of rows do not behave as expected on some platforms.** **Workaround**: For best performance, keep the number of rows in choice controls under 200.
CR 682741	**"Validate controls as soon as the user tries to change focus away from them" option in Mobile Workflow Package generation wizard is not supported on Windows Mobile.** **Workaround**: None.
CR 681918	**When you click the Workflow icon on iOS devices, you see the message `Sybase Mobile Workflow please go to the settings application and enter your configuration...`.** This happens if you attempt to open a version 2.0 workflow on the iOS device and connect to version 2.1 of Unwired Server. **Workaround**: In Sybase Control Center, use the HWC template to create an application connection and leave the application identifier empty.

Issue Number	Description
CR 678440	**On BlackBerry devices, you receive an error when running an HTTPS GET method from the Mobile Workflow** **Workaround:** Refer to the BlackBerry Knowledge Base article at *http://btsc.webapps.blackberry.com/btsc/search.do?cmd=displayKC&docType=kc&externalId=KB20833&sliceId=1&docTypeID=DT_SUPPORTISSUE_1_1&dialogID=1652002964&stateId=0%200%201652004660*.
CR 677684	**Locale properties file is not updated if screens, controls, menu items and so on are removed.** During Mobile Workflow package generation, you see a warning that validation keys are missing from the `<locale>.properties` file if screens, controls, menu items and so on have changed in number, for example, if you deleted a menu item. Resource strings associated with screens, controls, and menu items do not correctly update the sequence. **Workaround**: Either do not remove screens, controls, menu items and so on after localizing, or manually correct the localization `.properties` file after making the changes.
CR 675904	**Object queries in the Online cache group are case-sensitive.** If the enterprise information system (EIS) to which an object query is associated with has case-sensitive parameters, the client must pass matching parameters or no data is returned. For example, if the Customer MBO is in an Online cache group, has a load parameter "state" that is propagated to the attribute "state," and a findByParameter object query defined as: `SELECT x.* FROM Customer x WHERE x.state = :state` No data is returned to the client if they enter `state=Ca`, since the database parameter is case-sensitive (requiring "CA"). This is not an issue for object queries in other types of cache groups. **Workaround**: None.

Issue Number	Description

CR 674149	**Exception received with large amounts of data.**
	When the amount of data is too large to download to the workflow client device, you may get an exception similar to:
	`"iAnywhere.MobileOffice.AMP.ResponseRetryException" on device, and following error in WorkflowClient.log:`
	`20110616T110249.554|4|My:1 -- ExecuteRequest- > Out`
	`20110616T110249.694|1|iAnywhere.MobileOffice.AMP.ResponseRetryException: Exception of type 'iAnywhere.MobileOffice.AMP.ResponseRetryException' was thrown.`
	`20110616T110249.694|1| at Sybase.UnwiredPlatform.WorkflowClient.Utils.CheckMessageSize (String serializedMessage, ContextData oContextData, Boolean fromResponder)`
	`20110616T110249.694|1| at Sybase.UnwiredPlatform.WorkflowClient.Responder.ProcessMessageFromDevice (ContextData oContextData, String sData, Boolean isSynchronous)`
	`20110616T110249.694|1| at Sybase.UnwiredPlatform.WorkflowClient.Responder.ProcessSyncRequest (ContextData oContextData, String sData)`
	`20110616T110249.694|4|My:1 -- ProcessSyncRequest- > Out`
	Workaround: Increase the allowed maximum workflow message size (SupMaximumMessageLength). From Sybase Control Center, expand Workflows and select the workflow of interest. Select the Context Variables tab. Modify the property.

CR 673572	**Workflows do not work with Android simulators version 2.3.3.** See *http://code.google.com/p/android/issues/detail?id=12987* for a description of this issue. **Workaround**: Use an earlier version of the Android simulator.
CR 671242	**Submit operation is empty if the workflow uses a Web Service that takes a structure as its input.** While you can add items to the listview and save the list, when you submit the operation, it is empty. Array/list fields in structure parameters are not supported. You can have a top-level parameter that is an array/list, and you can have a top-level parameter that is a structure that has non-array/list fields and substructures. **Workaround**: None.

Issue Number	Description
CR 669251	**On a BlackBerry 9630 running OS 5.0, data with the DATETIME type does not display in certain circumstances.** **Workaround**: None.
CR 667325	**On BlackBerry 5.0, if there is a dynamic choice on the screen, labels for drop-down and comments fields overlap, which hides the drop-down control.** This happens only on some BlackBerry 5.0 devices, and only intermittently. **Workaround**: If you know where to click for the drop-down, you can access the drop-down control, and it accepts your choice.
CR 667187	**The BlackBerry Bold 9700 device simulator sometimes does not allow the user to have full control of the screen for an application, and may perform the wrong operations, and return the wrong results. Real devices work correctly.** **Workaround**: None.
CR 666672	**On BlackBerry 6.0 simulators (non-touch models), it is difficult to enter the password.** **Workaround:** There are two workarounds for this issue: **1.** When the password control has focus, press the scroll button and scroll to the left. **2.** When the password control has focus, press the Menu button, choose **Select**, then move the cursor to the password control.

Known Issues for Native Object API

Learn about known issues and apply workarounds for Native Object API and custom development.

Issue Number	Description
CR 710279	**Exception not thrown when failing to set personalization parameters or synchronization parameters.** When an application starts up for the first time, you should first perform an initial synchronization before setting personalization parameters and synchronization parameters. Without an initial synchronization, setting the personalization parameters and synchronization parameters fails. In that case, although the saving of the personalization parameters and synchronization parameters fails, no exception is thrown, as if these operations succeed. **Workaround:** Ensure that you perform an initial synchronization before setting personalization parameters and synchronization parameters.
CR 709534	**Synchronization notification does not mean that asynchronous operation replay is complete.** In an Android, BlackBerry, or Windows or Windows Mobile application, when you call `DBClass.beginSynchronize(synchronizationGroups,userContext,uploadOnly)` with `AsyncReplay=true`, the `userContext` is passed to the server and used in `CallbackHandler.onSynchronize()` with `SynchronizationStatus=ASYNC_REPLAY_COMPLETED`. If `CallbackHandler.onSynchronize()` returns `CONTINUE`, the Object API uses the same `userContext` for the triggered `synchronize`. **Workaround**: Call `beginSynchronize` manually from the `onSynchronize` callback and return `CANCEL` to disable the triggered callback.

Issue Number	Description
CR 709392	**Missing error code methods for the BlackBerry client in com.sybase.persistence.SynchronizationContext.** The BlackBerry client is missing the following Object API methods for getting and settings error codes and messages in `com.sybase.persistence.SynchronizationContext`: • `getErrorCode` • `setErrorCode` • `getErrorMessage` • `setErrorMessage` **Workaround:** Patch 01 fixes this issue. See the *Fixed Issues in Patch 01* in the *Sybase Unwired Platform 2.1 ESD #3 Patch 01* section of this release bulletin. If you cannot apply the patch, there is no workaround.

CR 709353	**Cannot pass cookie through Personalization Key in RBS.**
	Workaround: Use the following code:
	`SUPStringProperties *cookies = [smnwTests verizonCookies];`
	`[[[SUPApplication getInstance] connectionProperties] setHttpCookies:cookies];`

CR 709073	**Operation replay records may not be processed correctly.**
	In some cases, operation replay records that you upload to the Unwired Server are not processed in the correct order on the server. For example:
	`[customer create];  // inserts a row into the MBO table`
	`[customer submitPending];  // inserts a row into the operation replay table`
	`[customer delete];  // deletes the row from the MBO table (because of server-side processing, does not remove the row from the operation replay table)`
	`[customer submitPending];  // does nothing`
	Workaround: Call `cancelPending` operation to cancel any pending record before submitting another operation.

CR 708708	**Ultralite database is not automatically backed up on iOS devices.** Due to an Ultralite database limitation, the Ultralite database file is stored in the application Documents directory and the "do not backup" attribute of the file is set. The data is not automatically backed up. **Workaround**: If needed, perform a manual backup of the data.

Issue Number	Description
CR 708707	**Best practice for deleting the client database** Asynchronous operation replay is enabled by default. When the application is connected (by `Application.StartConnection()` or `Application.RegisterApplication`), the application may receive background notifications and will trigger a `synchronize` or other database operation. If you try to delete the database, you may receive database exceptions. **Workaround**: Before deleting the database, stop the application connection (`Application.StopConnection()`).

CR 708624	**Synchronization with Network Edge Auth SSO on BB device.**
	When synchronizing with Network Edge Auth SSO on BlackBerry device `UltraLiteJ Error[-85]: Communication error (HTTP 500, URL http://ucperflab01.sybase.com//rs/client/rs_client.dll/supnw.rep1` appears.
	Workaround 1: Set Synchronization Server URL Suffix of Application Connection Templates to `rs/client/rs_client.dll/{RBSFarmID}`.
	Workaround 2: Set the following line in your connection profiles:
	`ConnectionProfile sp = SSOWSDB.getSynchronizationProfile();`
	`NetworkStreamParams streamParams = sp.getStreamParams();`
	`streamParams.setUrl_Suffix("rs/client/rs_client.dll/{RBSFarmID}");`
	`sp.save(); ...`

CR 692374	**Enabling compression incurs additional processing on the device and the server side to optimize the amount of data to be transferred over the network**. **Workaround**: Carefully evaluate whether the compression trade-off is appropriate in the deployment environment, and carefully size the server capacity to support it. If appropriate in a particular environment, follow the Sybase Unwired Platform *Developer Guides* to enable compression.

Issue Number	Description

CR 690308	**The query.getRow() API may not return the correct row to BlackBerry clients.**
	For example, in this connected result set for a BlackBerry client:
	`Query query = new Query();`
	`query.select("c.id as Id, c.fname as Fname, c.state as countState");`
	`query.from("Customer", "c");`
	`query.setConnectedResultSetEnabled(true);`
	`QueryResultSet rs = MyDatabase.executeQuery(query);`
	`rs.executeQuery();`
	`rs.last();`
	`rs.afterLast();`
	`rs.previous();`
	`rs.getRow()` may not return the correct value.
	Workaround: Do not use the query.getRow() API QueryResultSet.previous\next return boolean to indicate if the current position is valid. Rely on the return value instead.

CR 675716	**SQLite cannot support create operations with more than 3MB of data, which is roughly 128 columns at 64KB per column on Windows Mobile devices.** In Windows Mobile applications that use message-based synchronization, when a row reaches 3MB or more, and the client invokes the SQLite create operation during subscription, an internal exception is generated, data import fails, and the import operation fails to commit the transaction. **Workaround**: None.

Known Issues for Hybrid Web Container

Learn about known issues and apply workarounds for Hybrid Web Container (HWC) development.

Issue Number	Description
CR 708778	**If the Relay Server credentials dialog is presented and the iOS Hybrid Web Container application is closed and re-opened, any previously saved credentials are retried automatically**. **Workaround**: Do not close the application when the credentials dialog is presented.

Issue Number	Description
CR 708173	**When the Hybrid Web Container tries to retrieve data from the MBO package an "Invalid application id <***> for package" error is received**. **Workaround:** In Sybase Control Center, manually add an Application ID for the MBO package's application list. **1.** Log in to Sybase Control Center. **2.** Go to **hostname@localhost > Applications** and click the **Applications** tab. **3.** Click **Properties**. **4.** In **Domains and Packages**, select the Domain that contains the applicable MBO package. **5.** Assign the Application ID to the MBO's package's Applications list.
CR 708122	**Native application on Windows Mobile device receives an error, "Can't find an Entry Point 'DLL_setMclCallbacks' in a PInvoke DLL 'CMessagingCLient.dll".** **Workaround**: This can occur if a 2.1.2 (or older) version of the Hybrid Web Container (HWC) is installed on the same device as a native application using the SUP 2.1.3 libraries. The HWC and native applications share a common CMessagingCLient.dll. In order for a 2.1.2 (or older) HWC and a 2.1.3 native app to coexist and both successfully run on the same Windows Mobile device, the 2.1.3 version of CMessagingCLient.dll should be the only one installed and loaded into memory on the device. Make sure older versions of that DLL are removed from the device prior to starting either application.

Issue Number	Description

CR 707616	**When writing a Hybrid Web Application for iOS using the PhoneGap API `navigator.camera.getPicture()` and specifying `FILE_URI` as the destination type, you cannot display the image**. The same is true for the result of any call to `FileEntry.toURI()`, `DirectoryEntry.toURI()`, or anything else that references file URIs.
	
	The problem occurs because of a security restriction that does not allow pages with non-local schemes from loading resources with local schemes. File URIs are considered local schemes.
	
	Workaround: Patch 01 fixes this issue. See the *Fixed Issues in Patch 01* in the *Sybase Unwired Platform 2.1 ESD #3 Patch 01* section of this release bulletin.
	
	If you cannot apply the patch, then use this workaround: Modify the URL so that it looks like a local file path without the extra file URI syntax. One way to do this is to intercept the callback at the javascript layer, and strip off the `file://localhost` or `file://` at the beginning of the URL. This leaves a bare file path like `/var/mobile/...`. Using the PhoneGap getPicture() function as an example, your success callback function might have this code:
	
	`function onPhotoURISuccess(imageURI) {`
	`var newURI = imageURI.replace("file://localhost","");`
	`newURI = newURI.replace("file://","");`
	`document.getElementById('MyImg').src = newURI;`
	`}`

Issue Number	Description

CR 705895	**After adding JavaScript code in the customAfterMenuItemClick method to update the checkbox (using JQuery Mobile) state on the screen, the checkbox fails to update**.
	
	Workaround: When you use JavaScript to update the checkbox (using JQueryMobile) value, you must also add the refresh code. For example:
	`Whole sample in "customAfterMenuItemClick" method of Custom.js`
	`function customAfterMenuItemClick(screen, menuItem) {`
	`  if (screen === "Start" && menuItem === "setCheckbox") {`
	`  var checkbox1 = document.getElementById('checkbox1');`
	`  checkbox1.checked = true;`
	`  var checkbox2 = document.getElementById('checkbox2');`
	`  checkbox2.checked = false;`
	`  if (isJQueryMobileLookAndFeel) {`
	`  $("input[type='checkbox']").checkboxradio("refresh");`
	`  }`
	`  }`
	`}`

CR 705219	When the iOS Hybrid Web Container application tries to retrieve seeding information from an Afaria iOS App store hosted package you receive an error message: `Afaria client application is installed but not configured. Please configure Afaria client application and try again or go to the Settings application and enter your configuration information.` **Workaround**: • Seeding works for Enterprise hosted packages on the Afaria server. • You can still manually configure the connection information.
CR 703883	**Switching from one edit box to another on a BlackBerry 7.0 simulator or device using the touch screen or mouse requires two clicks or touches.** **Workaround**: None for BlackBerry 7.0. Upgrading to BlackBerry 7.1 eliminates the problem.
CR 702603	**On BlackBerry 7.0 devices, a white screen is seen until the Workflow application's activation screen opens**. See *https://www.blackberry.com/jira/browse/JAVAAPI-119* **Workaround**: None.

Issue Number	Description
CR 693907	**Hybrid Web Container occasionally fails on Android 2.2.** This can happen when all of these conditions are present: • There is only one read-only edit box control on a single screen • Using jQueryMobile GA version • On an Android 2.2 simulator or Samsung Galaxy device with Android 2.2 **Workaround**: Either: • Upgrade the OS on the device to Android 2.3or • If using an Android 3.x simulator, see CR 673572. or • Add one or more controls to the screen
CR 693381	**Cannot preview a selected picture On Windows Mobile emulator.** When using the imageElement.src = imageUri in the getPicture function, you cannot preview the selected image on the Windows Mobile emulator. This is because the HTML browser adds an unexpected prefix to the URI, so the Hybrid Web Container cannot find the file to which the URI refers. **Workaround**: Either: • Preview the image using an actual device instead of the emulator, or • Assign image data instead of image URI to the image element so the image data can be retrieved through the JavaScript API
CR 691005	**Image upload may fail on BlackBerry 6 simulators older than version 6.0.0.499.** **Workaround**: Use a BlackBerry 9800 simulator with version 6.0.0.499 or later.

Issue Number	Description

CR 688399	**On Windows Mobile devices, thumbnails do not show properly if an image has a large size.**.
	
	Workaround: The root cause is that Internet Explorer cannot load a large image; try one of these solutions to address the image size:
	
	Solution 1: Specify a small-sized image as the thumbnail instead of using the original large image, for example:
	`<img src="default-thumbnail.jpg" width="120" height="120" id="Thumbnail" sup_static_options="true" name="Preview" alt="Preview"/>`
	
	Solution 2: Pass the width and height as the parameters in the source of the image control, so the container compresses the original image on the fly, for example:
	`<img src="default.jpg?width=120&height=120" width="120" height="120" id="Thumbnail" sup_static_options="true" name="Preview" alt="Preview"/>`
	
	The width and height parameters are supported only on Windows Mobile and may cause display issues on other platforms.

CR 686465	Camera feature is supported on some device simulators/emulators. • iOS: not supported. • Android 2.2: not supported. *http://code.google.com/p/android/issues/detail?id=9376* • BlackBerry: supported. • Windows Mobile: supported; however, can photograph only a grey/black screen. **Workaround**: Use a physical device.
CR 680253	**Images do not appear correctly on mobile devices.** Hybrid Web Container/Workflow applications can display pictures from binary enterprise information system (EIS) data. But when a MBO developer drags and drops a table that contains an image field to create the MBO, the image field maps to a Binary datatype that has a length of approximately 3KB, which may not be large enough for the entire image. At runtime, the binary data of the image may be truncated in the cache database (CDB), and not appear correctly on the device. **Workaround**: When creating the MBO, manually modify the length of the binary field to make sure it is large enough to hold the image binary data, or use the BigBinary datatype instead.

Known Issues for OData SDK

Learn about known issues and apply workarounds for the OData SDK.

Issue Number	Description
n/a	Certificate provisioning using Afaria APIs While using the Afaria APIs for certificate provisioning, the application developer must ensure that the common name passed to the function is verified. Afaria makes no attempt to verify the identity of the certificate requestor.

Known Issues for MAKit

Learn about known issues and apply workarounds for MAKit development.

Issue Number	Description
CR 733167	**When opening a MAKit PDF from the Product Documentation Web site, some users may be prompted to update Adobe Flash Player.** When some users attempt to open the MAKit Developer Guides at *http://sybooks.sybase.com/sybooks/sybooks.xhtml?id=1289&c=firsttab&a=0&p=categories*, they may get an error stating an updated version of Flash is required to view the PDF package. **Workaround**: Update to the most recent version of both Adobe Reader and Adobe Flash Player by visiting the Adobe Downloads Web site at *http://www.adobe.com/downloads/*.

CR 721724	**The MAKit 2.1.3 Code Example orientation freezes in Apple iOS 6.0 .**
	When attempting to change the orientation from landscape to portrait, or from portrait to landscape on the iPad2, the MAKit Code Example does not change orientation.
	Workaround: To run the MAKit 2.1.3 in iOS6.0 without any orientation issues:
	1. In `MAKitCodeExamplesAppDelegate` replace the line
	`[self.window addSubview:navController.view];`
	with
	`[set.window setRootViewController:navController];`
	2. In the `MAKitCodeExamples-info.plist` file, delete the sub items (item 0 and item 1) under the "Supported interface orientations".

Issue Number	Description
CR 704674	**Value bubble and range selector are not kept in sync.** When the user manipulates the range selector, the highlight bar of the value bubble and the category values are not kept in sync. **Workaround**: User can tap anywhere on the chart to synchronize the highlight bar of the value bubble and the category values.
CR 704130	**HTML5 - Pie chart series missing from legend box.** The issue occurs when the application: • Shows six pie charts on one page with pie chart having additional series. • Shows one pie chart with additional series and then the user maximizes browser window on the mobile phone. **Workaround**: None.
CR 703197	**SUPContainer iOS (iPad) what-if integer columns turns blank after clearing the value.** When the MAKit What-if feature is used in the Unwired Platform Workflow container environment on iOS (iPad devices), the `what-if` integer column becomes blank and the slider does not reposition when the value is cleared using device keyboard input and the user taps out of the field. This field should have a value of 0 after clearing and the slider control should be synchronized to the 0 position. **Workaround**: User can use the slider control to change the what-if column values.

Documentation Issues and Updates for Sybase Mobile SDK 2.1 ESD #3

Read about updates, corrections, and clarifications to the documentation released with Sybase Unwired Platform Mobile SDK.

Issue Number	Description
n/a	**Duplication linking issues in *Sybase Control Center for Sybase Unwired Platform*** In the *Applications* topic, there are dulicate links to these topics: • Application Creation • Application Connection Activation Options • Application Users • Application Connections • Application Connection Templates • Application Connection Properties There are duplicate Parent topic links back to the *Applications* topic in these topics: • Application Creation • Application Connection Activation Options • Application Users • Application Connections • Application Connection Templates • Application Connection Properties

n/a	**Close parentheses missing from linker flags**
	In the *Importing Libraries and Code* topic in *Developer Guide: iOS Object API Applications* step 11 should include close parentheses after `EFFECTIVE_PLATFORM_NAME`:
	`$(SRCROOT)/$(PRODUCT_NAME)/libs/$(CONFIGURATION)$(EFFECTIVE_PLATFORM_NAME)/libMO.a`
	`$(SRCROOT)/$(PRODUCT_NAME)/libs/$(CONFIGURATION)$(EFFECTIVE_PLATFORM_NAME)/libSUPObj.a`

n/a	**Corrected code for setting up a database connection** The code in *Connecting Through a Relay Server* on page 59 was updated to cover two situations: whether the connection template on SCC is configured with all the required Relay Server information or not.

Issue Number	Description
n/a	**On-demand cache policy with a small coherence window, not large** In the *Cache Group Considerations* topic in *Mobile Data Models: Using Mobile Business Objects 2.1 ESD #3*, the second sentence should read: "...for transactional data that has little tolerance for EIS data that is stale, an On-demand cache policy with a small coherence window is a more appropriate cache solution than an hourly schedule-based refresh."
n/a	**Do not compile and run the example projects** The following note was omitted from the prerequisites in the Unwired Platform version 2.1 ESD #3 Tutorials: Do not try to compile and run the tutorial example projects posted in the SAP® Community Network: *http://scn.sap.com/docs/DOC-8803*. They are meant only as code examples. This applies to: • *Tutorial: Android Object API Application Development* • *Tutorial: BlackBerry Object API Application Development* • *Tutorial: iOS Object API Application Development* • *Tutorial: Windows Mobile Object API Application Development* • *Tutorial: Mobile Workflow Package Development*
n/a	**Updates to the Datavault documentation for Android were made.** A new topic was added in *Developer Guide: Android Object API Applications > Client Object API Usage > Security APIs > `SUPDataVault` DataVault* to document the `init` method. *Developer Guide: Android Object API Applications > Client Object API Usage > Security APIs > `SUPDataVault` DataVault > Code Sample* is also updated to include the `init` method.
n/a	**Changes in the directory structure for the Sybase Mobile SDK installation in this version were not updated in the documentation.** Beginning with Unwired Platform version 2.1 ESD #3, Sybase Mobile SDK installs in a directory structure that allows multiple versions of Mobile SDK to coexist on the same system. The SDK top level directory now includes the version and some directories that were peers to the SDK top level directory are now located below it. Refer to *Path Corrections for Backward Compatibility* on page 152 when you encounter a path specifying the location of any Mobile SDK files.

Issue Number	Description
n/a	In the *Developer Guide: Mobile Workflow Packages > Mobile Workflow Development > Hybrid Web Container Customization > iOS Hybrid Web Container Customization > iOS Customization Touch Points > Default Behavior Customization for the iOS Hybrid Web Container > Customizing PIN Screens on iOS > Removing the PIN Screen*, there is additional information to note: **Note:** Removing the PIN screen leaves data that is stored on the device less secure. You should remove the PIN screen only if you are not concerned about keeping your data secure. This also applies to *Developer Guide: Mobile Workflow Packages > Mobile Workflow Development > Hybrid Web Container Customization > Android Hybrid Web Container Customization > Android Customization Touch Points > Default Behavior Customization for the Android Hybrid Web Container > Removing the PIN Screen*.

SMPONP-3383	**Documentation does not mention need to re-get sync parameter instance**
	In *Developer Guide: iOS Object API Applications 2.1 ESD #3 > Customizing the Application Using the Object API > Initializing an Application > Initially Starting an Application > Modifying Synchronization Parameters*, when modifying synchronization parameters, you must re-get the sync parameter instance and save again.
	Workaround: Use the correct code:
	`<MBO>SynchronizationParameters *params = [<MBO> getSynchronizationParameters];`
	`[params delete];`
	`params = [<MBO> getSynchronizationParameters]; //must re-get the sync parameter instance`
	`params.Param1 = @value1; //set new sync parameter value`
	`params.Param2 = @value2; //set new sync parameter value`
	`[params save];`
	`[SUP101SUP101DB synchronize];`

Issue Number	Description
CR 724568	**Use submitPendingOperations and cancelPendingOperations only when there are multiple pending entities on the same MBO type.** **submitPendingOperations** and **cancelPendingOperations** do a select on the MBO table to find the pending MBO instances. If the MBO instance is available in memory, using the MBO instance's **submitPending** or **cancelPending** is more efficient. Applies to the *Pending Operation* topic in the Object API developer guides: • Android Developer Guide • BlackBerry Developer Guide • iOS Developer Guide • Windows Mobile Developer Guide • Windows Developer Guide
CR 721074	**Topic no longer applicable in the BlackBerry developer guide** The *HTTP Push Gateway* topic in *Developer Guide: BlackBerry Object API Applications > Client Object API Usage > Synchronization APIs > Push Synchronization Applications* is not applicable in Unwired Platform version 2.1 ESD #3.
CR 716173	**MBS is used for authentication** The topic *Object API Changes in SDK Version 2.1 ESD #3* in *New Features 2.1 ESD #3 > New Features for Sybase Mobile SDK > Object API Development* requires clarification. The sentence "Consequently, messaging-based synchronization has been removed from the 2.1 ESD #3 SDK." should be changed to: "Consequently, messaging-based synchronization has been removed from the 2.1 ESD #3 SDK for data delivery. The messaging channel continues to be used for communications such as application registration." The types of communications the messaging channel sends are listed in *DeveloperGuide: iOS Object API Applications > Getting Started with iOS Development > Best USes for Object API Applications > Client Runtime Architecture > Mobile Channel Interfaces*.

Issue Number	Description

CR 714445	References in the *Developer Guide: Mobile Workflow Packages* > *Mobile Workflow DCN Without Payload* topic are not valid JSON, because the data value is an object and should be an array, and there is an extra colon. For example, this syntax is incorrect:
	`{"id":"","op":":","subject":"","to":"","from":"","read":,"priority":,"body":"", "data{}}`
	And should be:
	`{"id":"","op":":upsert","subject":"","to":"","from":"", "read":,"priority":, "body":"","data":[]}`
	And this example:
	`{"id":"","op":":","subject":"test", "to":"test","from":"test","read":,"priority":,"body":" MATCH:SUP_MWF,TaskID:TS97200149, WIID:1470577, USER:PERF0111*#END#*", "data"{}}`
	Should be:
	`{"id":"","op":":upsert","subject":"test","to":"test","from":"test","read":, "priority":,"body":" MATCH:SUP_MWF,TaskID:TS97200149, WIID:1470577, USER:PERF0111*#END#*","data":[]}`

CR 710930	**Startup of an application always gets http callback errors**. During initialization, the client may try several URLs in an attempt to discover the correct path to the Unwired Platform server. During these attempts, 404 errors (Page Not Found) may occur and will be reported to the OnHttpCommunicationError callback. This is expected behavior and you can safely ignore these initial 404 errors. If the 404 errors continue to occur after successful registration of the application or if registration never succeeds, double check the UrlSuffix setting of the application's ConnectionProperties object.
CR 710608	**Return code is unexpected through onHttpCommunicationError with disabled user**. In an environment configured for SiteMinder authentication, the expectation is that onHttpCommunicationError will be called back for error condition 302. This condition is not fulfilled currently by the Windows and Windows Mobile platforms; instead, they return 404 errors.

Issue Number	Description
CR 710105	**Issue:** Unwired WorkSpace online help for output mapping does not link to a valid topic. Search Sybase Unwired Platform documentation for the topic Output Mappings for more information.
CR 709826	**Clarification on onReplaySuccess:(id)entityObject.** In the *Developer Guide: iOS Object API Applications > Client Object API Usage > Callback and Listener APIs*, there is additional information to note: **Note:** The `onReplaySuccess:(id)entityObject` is an MBO object instance that contains the data prior to the synchronization. You can use the change log API to find records that occur after the synchronization.
CR 709815	**Issue**: The synchronization method described in the topic Synchronization Parameter Behavior in the Mobile Data Models: Using Mobile Business Objects guide is incorrect. References to sp.Synchronize() should be SUP101DB.Synchronize().
CR 709519	**Issue**: Some documents reference an incorrect path: `<Install-Root>\MobileSDK213\ObjectAPI\iOS\MBS\iOS` and `<Install-Root>\MobileSDK213\ObjectAPI\iOS\RBS\iOS`. iOS client files for 2.1.3 are in `<Install-Root>\MobileSDK213\ObjectAPI\iOS\RBS` and `<Install-Root>\MobileSDK213\ObjectAPI\iOS\MBS`.
CR 709194	**Issue**: For all Object API tutorials, if you opt to use the Mobile Business Object example project instead of performing the tutorial, the instructions should include that you must deploy the mobile application project to Unwired Server. Creating the MBOs and deploying the project are prerequisites for performing the Object API tutorials. After importing the MBO example project, connect to Unwired Server and deploy the project. See *Tutorial: Mobile Business Object Development > Deploying the Database Mobile Business Objects* for details.
CR 706677	**Creating a DataVault that Enforces Password Policy** The data vault security feature has been enhanced to allow administrators to create a password policy for data vault logins. Developers must create data vault code that performs policy enforcement. Use the datavault class to create a datavault that enforces a password policy. This class provides encrypted storage of occasionally used, small pieces of data, and is documented in the SUPDataVault or DataVault topic in the corresponding Developer Guide for your application type.

Issue Number	Description
CR 708259	**The developer guides do not document all the methods in the ConnectionPropertyType class.** See the generated API reference information provided with the Mobile SDK for a complete list of methods in the `ConnectionPropertyType` class.
CR 674889	**Apple Push Notification Services (APNS) Works on iPad Devices.** The *Apple Push Notification Service Configuration* topic in the *OData SDK Developer Guide* and the *iOS Object API Applications Developer Guide* incorrectly states that APNS cannot be used with iPad devices.

Mobile Channel Interfaces

Two main channel interfaces provide notifications and data transport to and from remote devices.

- The messaging channel serves as the abstraction to all device-side notifications (BlackBerry Enterprise Service, Apple Push Notification Service, and others) so that when changes to back-end data occur, devices can be notified of changes relevant for their application and configuration.

  The messaging channel sends these types of communications:
  - Application registration - the messaging channel is used for application registration before establishing a connection to the Unwired Server.
  - Change notifications - when the Unwired Server detects changes in the back-end EIS, the Unwired Server can send a notification to the device. By default, sending change notifications is disabled, but you can enable sending change notifications per synchronization group.

    To capture change notifications, you can register an `onSynchronize` callback. The synchronization context in the callback has a status you can retrieve.
  - Operation replay records - when synchronizing, these records are sent to the Unwired Server and the messaging channel sends a notification of `replayFinished`. The application must call another synchronize method to retrieve the result.
  - SAP Data Orchestration Engine (DOE) application synchronization - the messaging channel is used for synchronization for DOE applications.
- The synchronization channel sends data to keep the Unwired Server and client synchronized. The synchronization is bi-directional.

init

Initialization function that you must call with the application's context before you call any of the other vault methods. In addition to saving the context for later use, this method also initializes static member variables (such as encryption objects).

Syntax

```
public static void init(android.content.Context oContext)
```

Parameters

- **oContext –**
  `Valid application context.`

Returns

None.

Examples

- **Initialize**

```
DataVault.init(oContext);
```

Code Sample

Create a data vault for encrypted storage of application data.

```
public void testFunctionality(Context oContext)
  {
    try
    {
       DataVault oDataVault = null;

       DataVault.init( oContext );

       // If this dataVault already exists, then get it by calling
getVault()
       // Else create this new dataVault by calling createVault()
       if ( DataVault.vaultExists( "DataVaultExample" ) )
         oDataVault = DataVault.getVault( "DataVaultExample" );
       else
         oDataVault = DataVault.createVault( "DataVaultExample",
"password!1A", "saltD#ddg#k05%gnd[!1A" );

      // Call setLockTimeout(). This allows you to set the timeout of
the vault in seconds
       oDataVault.setLockTimeout( 1500 );
       int iTimeout = oDataVault.getLockTimeout();

      // Call setRetryLimit(). This allows you to set the number of
retries before the vault is destroyed
       oDataVault.setRetryLimit( 10 );
       int iRetryLimit = oDataVault.getRetryLimit();

      // Call setPasswordPolicy(). The passwordPolicy also includes
the retryLimit and LockTimeout that we set above.
       DataVault.DVPasswordPolicy oPasswordPolicy = new
DataVault.DVPasswordPolicy();
       oPasswordPolicy.setIsDefaultPasswordAllowed (true);
       oPasswordPolicy.setMinLength( 4 );
       oPasswordPolicy.setHasDigits( true );
       oPasswordPolicy.setHasUpper( true );
       oPasswordPolicy.setHasLower( true );
```

```
       oPasswordPolicy.setHasSpecial( true );
       oPasswordPolicy.setExpirationDays( 20 );
       oPasswordPolicy.setMinUniqueChars( 3 );
       oPasswordPolicy.setLockTimeout( 1600 );
       oPasswordPolicy.setRetryLimit( 20 );

      // SetPasswordPolicy() will always lock the vault to ensure the
old password
       // conforms to the new password policy settings.
       oDataVault.setPasswordPolicy( oPasswordPolicy );

      // We are now locked and need to unlock before we can access the
vault.
       oDataVault.unlock( "password!1A", "saltD#ddg#k05%gnd[!1A" );

       // Call getPasswordPolicy() to return the current password
policy settings.
       DataVault.DVPasswordPolicy oCurrentPolicy =
oDataVault.getPasswordPolicy();

      // Call setString() by giving it a name:value pair to encrypt
and persist
       // a string data type within your dataVault.
       oDataVault.setString( "stringName", "stringValue" );

      // Call getString to retrieve the string we just stored in our
data vault!
       String storedStringValue =
oDataVault.getString( "stringName" );

       // Call setValue() by giving it a name:value pair to encrypt
and persist
       // a binary data type within your dataVault.
       byte[] binaryValue = { 1, 2, 3, 4, 5, 6, 7  };
       oDataVault.setValue( "binaryName", binaryValue );

      // Call getValue to retrieve the binary we just stored in our
data vault!
       byte[] storedBinaryValue =
oDataVault.getValue( "binaryName" );

      // Call getDataNames to retrieve all stored element names from
our data vault.
       DataVault.DVDataName[] dataNameArray =
oDataVault.getDataNames();
       for ( int i = 0; i < dataNameArray.length; i++ )
       {
          if ( dataNameArray[i].getType() ==
DataVault.DV_DATA_TYPE_STRING )
          {
             String thisStringValue =
oDataVault.getString( dataNameArray[i].getName() );
          }
          else
          {
             byte[] thisBinaryValue =
```

```
oDataVault.getValue( dataNameArray[i].getName() );
           }
       }

       // Call changePassword with 2 parameters. Vault must be
unlocked.
      // If you pass null parameters as your new password or your new
salt,
       // it will generate a default password or default salt,
respectively.
       oDataVault.changePassword( null, null );

       // Call isDefaultPasswordused() to see if we are using an
automatically
       // generated password (which we are).
       boolean isDefaultPasswordUsed =
oDataVault.isDefaultPasswordUsed();

       // Lock the vault.
       oDataVault.lock();

      // Call changePassword with 4 parameters even if the vault is
locked.
      // Here, we pass null for oldSalt and oldPassword because
defaults were used.
       oDataVault.changePassword( null, null, "password!1A",
"saltD#ddg#k05%gnd[!1A" );

       // Call isDefaultPasswordused() and we will see that the
default password is NOT used anymore.
       isDefaultPasswordUsed = oDataVault.isDefaultPasswordUsed();
    }
    catch( Throwable exception )
    {
      exception.printStackTrace();
    }
    finally
    {
      try
       {
        // Because this is a test example, we will delete our vault at
the end.
         // This means we will forever lose all data we persisted in
our data vault.
          if ( DataVault.vaultExists( "DataVaultExample" ) )
            DataVault.deleteVault( "DataVaultExample" );
       }
       catch(Throwable t)
       {
         t.printStackTrace();
       }
    }
  }
```

Creating a DataVault that Enforces Password Policy

The data vault security feature has been enhanced to allow administrators to create a password policy for data vault logins. Developers must create data vault code that performs policy enforcement.

Use the data vault class to create a data vault. This class provides encrypted storage of occasionally used, small pieces of data, and is documented in the *SUPData Vault* or *Data Vault* topic in the corresponding *Developer Guide* for your application type. Use the methods documented for the class to create the data vault, then refer to *Transferring Password Policy Values into the Data Vault* to learn how to enforce the password policy created by the administrator. For information about setting password policies, see *Enabling and Configuring a Password Policy for Data Vault Logins*.

Transferring Password Policy Values into the Data Vault

To use the password policy that you configured on Unwired Server, update your client application to transfer the password policy settings into the data vault.

Prerequisites

Create a data vault using a temporary password and salt. Create and save the encryption keys for the vault.

Task

Include the code similar to this C# example in your application. This code transfers the policy settings to the `PasswordPolicy` object, sets the policy on the data vault, then sets a new password that conforms to the policy. This example assumes a valid password already exists. In a real application, you would display a password dialog with text describing the policy details (such as minimum number of characters, uppercase/lowercase requirements, special characters allowed/not allowed), and have the user enter the password a second time for confirmation.

```
ConnectionProfile cp = SUP101DB.GetConnectionProfile();
DataVault vault = null;
// handle first-run initialization - create vault, set password
policy, set database key
if (!DataVault.VaultExists("myVault"))
{
  vault = DataVault.CreateVault("myVault", null, null);
  SUP101DB.GenerateEncryptionKey();
  vault.Unlock(null, null);
  vault.SetString("dbKey", cp.EncryptionKey);
  ApplicationSettings aps = app.ApplicationSettings;

  if (aps.IsApplicationSettingsAvailable())
  {
    bool policyEnabled = (bool)
aps.GetBooleanProperty(ConnectionPropertyType.PwdPolicy_Enabled);
    if (policyEnabled)
    {
```

```
      try
      {
        DataVault.PasswordPolicy oPasswordPolicy = new
DataVault.PasswordPolicy();
          oPasswordPolicy.defaultPasswordAllowed =
             (bool) aps.GetBooleanProperty(

ConnectionPropertyType.PwdPolicy_Default_Password_Allowed);
          oPasswordPolicy.minimumLength =
             (int) aps.GetIntegerProperty(
                ConnectionPropertyType.PwdPolicy_Length);
          oPasswordPolicy.hasDigits =
             (bool) aps.GetBooleanProperty(
                ConnectionPropertyType.PwdPolicy_Has_Digits);
          oPasswordPolicy.hasUpper =
             (bool) aps.GetBooleanProperty(
                ConnectionPropertyType.PwdPolicy_Has_Upper);
          oPasswordPolicy.hasLower =
              (bool) aps.GetBooleanProperty(
                 ConnectionPropertyType.PwdPolicy_Has_Lower);
          oPasswordPolicy.hasSpecial =
               (bool) aps.GetBooleanProperty(
                  ConnectionPropertyType.PwdPolicy_Has_Special);
          oPasswordPolicy.expirationDays =
               (int) aps.GetIntegerProperty(
               ConnectionPropertyType.PwdPolicy_Expires_In_N_Days);
          oPasswordPolicy.minUniqueChars =
                (int) aps.GetIntegerProperty(
                ConnectionPropertyType.PwdPolicy_Min_Unique_Chars);
          oPasswordPolicy.lockTimeout =
                (int) aps.GetIntegerProperty(
                   ConnectionPropertyType.PwdPolicy_Lock_Timeout);
          oPasswordPolicy.retryLimit =
                (int) aps.GetIntegerProperty(
                   ConnectionPropertyType.PwdPolicy_Retry_Limit);
        // SetPasswordPolicy() will always lock the vault to ensure
the old password
          // conforms to the new password policy settings.
          vault.SetPasswordPolicy(oPasswordPolicy);
          vault.ChangePassword(null, null, pwd, null);
       }
       catch (DataVaultException dve)
       {
         Console.WriteLine("password not good enough? " + dve);
       }
     }
   }
   vault.Lock();
 }
 else
 {
   vault = DataVault.GetVault("myVault");
 }

 if (vault.IsDefaultPasswordUsed())
 {
```

```
  vault.Unlock(null, null);
}
else
{
  vault.Unlock(pwd, null);
}
cp.EncryptionKey = vault.GetString("dbKey");
```

Path Corrections for Backward Compatibility

Changes in the directory structure for the Sybase Mobile SDK installation in this version, which were not updated in the documentation, are summarized here.

When you encounter a path in the product documentation for Unwired Platform version 2.1 ESD #3 that includes any of the following directories, refer to this topic to see the path change in all cases:

- Eclipse
- HybridWeb
- JDK1.6.X_XX
- MAKit
- MobileSDK
- OData
- ObjectAPI
- sapjco
- Unwired_WorkSpace

When you encounter a path in the product documentation for Unwired Platform version 2.1 ESD #3 that includes any of the following directories, refer to this topic to see the path change if the text is referring to Sybase Mobile SDK:

- -jvm
- InstallLogs
- ThirdParty
- Uninstallers

There are three categories of path changes for Mobile SDK. The following sections describe these categories and list the SDK folders affected.

Mobile SDK Folder Name Changes to "MobileSDK213"

Mobile SDK files that were under the Mobile SDK folder in earlier versions are still under the Mobile SDK folder in 2.1.3, but that folder name now incorporates the version number. The directory structure under these folders does not change.

Path to Earlier Version SDK Files	Path to 2.1.3 SDK Files
MobileSDK_InstallDir \MobileSDK\HybridWeb	*MobileSDK_InstallDir* \MobileSDK213\HybridWeb

Path to Earlier Version SDK Files	Path to 2.1.3 SDK Files
MobileSDK_InstallDir \MobileSDK\MAKit	*MobileSDK_InstallDir* \MobileSDK213\MAKit
MobileSDK_InstallDir \MobileSDK\ObjectAPI	*MobileSDK_InstallDir* \MobileSDK213\ObjectAPI
MobileSDK_InstallDir \MobileSDK\OData	*MobileSDK_InstallDir* \MobileSDK213\OData

Some Folders that Were Siblings to Mobile SDK Folder Become Children
Mobile SDK folders that were siblings of the Mobile SDK folder in earlier versions are now under the renamed Mobile SDK folder.

Path to Earlier Version SDK Files	Path to 2.1.3 SDK Files
MobileSDK_InstallDir \Eclipse	*MobileSDK_InstallDir* \MobileSDK213\Eclipse
MobileSDK_InstallDir \JDK1.6.0_31	*MobileSDK_InstallDir* \MobileSDK213\JDK1.6.0_31
MobileSDK_InstallDir \MobileSDK\sapjco	*MobileSDK_InstallDir* \MobileSDK213\sapjco
MobileSDK_InstallDir \MobileSDK\Unwired_WorkSpace	*MobileSDK_InstallDir* \MobileSDK213\Unwired_WorkSpace

Other Folders that Were Siblings to Mobile SDK Folder May Now Be Duplicated
In earlier versions of Unwired Platform, some folders that were siblings of the Mobile SDK folder contained files for Runtime and Mobile SDK. In this version, if Mobile SDK is installed on a system where Unwired Platform Runtime is installed, the Runtime files remain in those folders as siblings of the Mobile SDK folder, but the SDK files move into a folder with the same or similar name under the 2.1.3 Mobile SDK folder.

If the documentation refers to Mobile SDK files, you must make the path changes detailed below. If the documentation refers to Runtime files, do not change the specified path.

Path to Earlier Version SDK Files	Path to 2.1.3 SDK Files
UnwiredPlatform_InstallDir \UnwiredPlatform\-jvm	*MobileSDK_InstallDir*\MobileSDK213\-jvm
UnwiredPlatform_InstallDir \UnwiredPlatform\InstallLogs	*MobileSDK_InstallDir*\MobileSDK213\InstallLogs

Path to Earlier Version SDK Files	Path to 2.1.3 SDK Files
UnwiredPlatform_InstallDir\UnwiredPlatform\ThirdParty	*MobileSDK_InstallDir*\MobileSDK213\ThirdParty
UnwiredPlatform_InstallDir\UnwiredPlatform\Uninstallers	*MobileSDK_InstallDir*\MobileSDK213\Uninstaller

Obtaining Help and Additional Information

Use the Product Documentation site and online help to learn more about this product release.

- Product Documentation at *http://sybooks.sybase.com/* – online documentation that you can access using a standard Web browser. You can browse documents online, or download them as PDFs. The Web site also has links to other resources, such as white papers, community forums, maintenance releases, and support content.
- Online help in the product, if available.

To read or print PDF documents, you need Adobe Acrobat Reader, which is available as a free download from the *Adobe* Web site.

Note: A more recent release bulletin, with critical product or document information added after the product release, may be available from the Product Documentation Web site.

Technical Support

Get support for SAP® products.

If your organization has purchased a support contract for this product, then one or more of your colleagues is designated as an authorized support contact. If you have any questions, or if you need assistance during the installation process, ask a designated person to contact Technical Support as specified in your contract:

- Sybase Technical Support or the Sybase subsidiary in your area
- SAP Technical Support

Customers with an SAP support contract can obtain support for this product at the SAP support site, *http://service.sap.com/sybase/support*. You can also find information about Sybase support migration at this location (login may be required).

Customers with a Sybase support contract can obtain support for this product at *http://www.sybase.com/support* (login required).

Downloading Product Updates

Get maintenance releases, support packages and patches, and related information.

- If you purchased the product directly from Sybase or from an authorized Sybase reseller:
  a) Go to *http://www.sybase.com/support*.
  b) Select **Support > EBFs/Maintenance**.
  c) If prompted, enter your MySybase user name and password.
  d) (Optional) Select a filter, a time frame, or both, and click **Go**.
  e) Select a product.

     Padlock icons indicate that you do not have download authorization for certain EBF/Maintenance releases because you are not registered as an authorized support contact. If you have not registered, but have valid information provided by your Sybase representative or through your support contract, click **My Account** to add the "Technical Support Contact" role to your MySybase profile.
  f) Click the **Info** icon to display the EBF/Maintenance report, or click the product description to download the software.
- If you ordered your product under an SAP contract:
  a) Go to *http://service.sap.com/swdc* and log in if prompted.
  b) Select **Search for Software Downloads** and enter the name of your product. Click **Search**.

Product and Component Certifications

Certification reports verify Sybase product performance on a particular platform.

To find the latest information about certifications:

- For partner product certifications, go to *http://www.sybase.com/detail_list?id=9784*
- For platform certifications, go to *http://certification.sybase.com/ucr/search.do*

Accessibility Features

Accessibility ensures access to electronic information for all users, including those with disabilities.

Documentation for this product is available in an HTML version that is designed for accessibility.

Vision impaired users can navigate through the online document with an adaptive technology such as a screen reader, or view it with a screen enlarger.

HTML documentation has been tested for compliance with accessibility requirements of Section 508 of the U.S Rehabilitation Act. Documents that comply with Section 508 generally also meet non-U.S. accessibility guidelines, such as the World Wide Web Consortium (W3C) guidelines for Web sites.

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

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

You may find additional information about accessibility features in the product documentation.