



Troubleshooting

Sybase Unwired Platform 2.0

DOCUMENT ID: DC01206-01-0200-01

LAST REVISED: May 2011

Copyright © 2011 by Sybase, Inc. All rights reserved.

This publication pertains to Sybase software and to any subsequent release until otherwise indicated in new editions or technical notes. Information in this document is subject to change without notice. The software described herein is furnished under a license agreement, and it may be used or copied only in accordance with the terms of that agreement.

To order additional documents, U.S. and Canadian customers should call Customer Fulfillment at (800) 685-8225, fax (617) 229-9845.

Customers in other countries with a U.S. license agreement may contact Customer Fulfillment via the above fax number. All other international customers should contact their Sybase subsidiary or local distributor. Upgrades are provided only at regularly scheduled software release dates. No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of Sybase, Inc.

Sybase trademarks can be viewed at the Sybase trademarks page at <http://www.sybase.com/detail?id=1011207>. Sybase and the marks listed are trademarks of Sybase, Inc. ® indicates registration in the United States of America.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world.

Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

Unicode and the Unicode Logo are registered trademarks of Unicode, Inc.

All other company and product names mentioned may be trademarks of the respective companies with which they are associated.

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

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

Contents

Introduction to Troubleshooting	1
Documentation Roadmap for Unwired Platform	1
Troubleshooting Basics	5
Collecting Information	5
Sybase Control Center Console Messages	5
Log Files	7
Reporting Errors	11
Reproducing Problems	11
Sybase Technical Support Checklist	12
Sending Error Log Fragments	12
Ensuring System Recoverability	13
Troubleshoot Basic Scenarios	15
Sybase Unwired Platform End-to-End Process Flow . . .	15
Debugging Message-Based Synchronization	17
Debugging Replication-Based Synchronization	20
Debugging the Cache	22
Debugging Data Change Notification	23
Debugging BlackBerry Device Development	26
Debugging iOS Device Development	28
Debugging Windows and Windows Mobile Device Development	30
Debugging DOE-C End-to-End Process Flow	33
Debugging JCO End-to-End Process Flow	34
Troubleshoot the System	37
Troubleshoot the Installation	37
Troubleshooting Host Name Problems	37
Listeners or Services Could Not be Started Errors	38
Unwired Platform Windows Services	43
Cannot Access Device Tab and Web Service Error	45

OpenDS Is Not Uninstalled	45
Error deleting <filepath>\APIdocs skipping unwired server update	45
Client Application Hangs Waiting for Authentication	46
Unwired Server or RSOE Startup Problems	48
Troubleshoot Deployment	49
Filter Class Files Not Copied to Deployment Target	49
Troubleshoot Synchronization	49
Synchronization Failed while Downloading Data	49
Multi-threaded Delete Failures Not Logged	50
Device Synchronization Returns no Data	50
Errors When Reusing Primary Keys	50
Troubleshoot the CDB Cache	51
Primary Key Record Missing	51
Data Caching Problems	51
Troubleshoot Security, Logins, and Authentication	52
Cannot Log in to Unwired Server	52
User Permissions Are Not Correct	52
Troubleshooting Single Sign-On	53
Troubleshoot EIS Connections	54
DOE-C or Messaging Connection Test Errors	54
Data Source Performance Errors	54
SAP Issues	55
Troubleshoot Servers	57
Unwired Server	57
Relay Server	60
Messaging Server	62
Troubleshoot Clusters	63
Advantage Database Server Cannot Authenticate	63
Troubleshoot Workflow DCN Failures	64

Troubleshoot Sybase Control Center for Sybase	
Unwired Platform	65
Using Sybase Control Center to Troubleshoot Unwired Platform	65
Collecting Administration Performance Data for Troubleshooting	66
Sybase Control Center Management Tier Issues	67
Launching SCC Results in Rounded Rectangle Box or Empty Console Screen	67
Sybase Unified Agent Windows Service Fails to Start	68
Sybase Unified Agent Windows Service Deleted	69
Sybase Control Center Fails to Start	69
Second Sybase Control Center Fails to Start	70
Login Invalid in Sybase Control Center	70
Cannot Access Unwired Platform Administration Nodes	71
Previous Administrator Credentials Used	71
Browser Refresh (F5) Causes Logout	71
Security Error Triggered When Connecting to SCC from Remote Browser	72
Administrator Login Passes When Provider Is Not Available	72
Host Name of Registered Resource Changed But Is Not Updated	72
Management Issues with Clustered Data Tiers	72
Platform Component Monitoring Issues	72
Monitoring Data Does Not Appear in History Tab	73
Previously Existing Monitoring Data No Longer Appears	73
Server Tier Administration Issues	74
Server List Not Retrieved	74
Unwired Server Fails to Start	76

Error in Listing Devices and ADMIN_WEBSERVICE_INVOCATION_ERR OR in Agent.log	76
Starting or Restarting a Remote Server from Sybase Control Center Fails	77
Port Conflict Issues	78
Unexpected Listener Startup or Connection Errors	78
Refreshing Server Configuration Displays Only Partial Updates	79
Users Connect with Old Credentials	80
AuthorizationException Displays Instead of Status	80
Increasing Messaging Queue Counts Degrades Performance	81
Saving Server Configuration Fails Due to Certificate Validation Error	81
Package Deployment and Management Issues	81
Exporting or Deploying Large Packages Fails	81
Invalid DOE-C User Error for an SAP Server Connection	82
Device and Device User Management Issues	83
Wrong Device for Code Error	83
User Name of Registered Device Not Displayed	84
Internal Server Error When Clicking Device User	84
Troubleshoot Mobile Business Object Development	87
Cannot Ping Unwired Server	87
Cannot Access sampledb	87
Data Source Connection Fails After Installing to a New Directory	88
MS SQL Server Connection Errors	88
ODBC Driver Errors	88
Mobile Business Object Errors	89

Troubleshooting MBOs	89
No Data or Images Appear on the Device	89
Troubleshooting Improperly Designed Mobile Business Objects	90
Database MBO Fails on Execution	91
Web Service MBO Sync Exceptions After an Interface Change	91
Binding an MBO to a Database Fails	91
Structure MBOs Not Automatically Generated After Rebinding	91
Multilevel Insert Operation Fails	92
Search Errors and the RTRIM Function	92
Update Operation Fails	92
Relationship Error	93
SQL Statement Yields Unexpected Results	93
Warning Message When Creating Relationships	93
Synchronization Fails if Binary Attributes Have Unspecified Length	94
Preview Errors in Unwired WorkSpace	94
Preview Errors for a Database Mobile Business Object	95
Preview Errors for a Web Service Mobile Business Object	95
Compiler Errors When Deploying Device Applications	95
Deploying to Unwired Server Fails	96
Deploying to Unwired Server in a Cluster Fails	96
Modified Port Changes Not Deployed	97
Troubleshoot Code Generation	99
SQL_MAX_ROW_SIZE_EXCEEDED	99
Error When Running Generated 32-bit Windows Code on a 64-bit Machine	99
Namespace Name Does Not Exist Error	100

Windows Mobile Code Generation Fails During Compilation	100
Cannot Generate Code with Javadoc Option Enabled	101
Code Generation Fails	101
Troubleshoot Custom Development	103
Insert Operation Fails for iOS	103
MBO Update Fails	103
Windows Mobile Compile Error For Generated API Project	104
Cannot Insert DEFAULT in Timestamp Column	105
Synchronization to SAP Data Source Repeats Excessively	105
Troubleshoot Mobile Workflow Forms Editor	107
Operation Fails with NULL Values Not Supported Error	107
Operation Fails with MinCheckAffectedRowCountException Error	107
Mobile Workflow Is Not in the Client List	108
Mobile Workflow Does Not Receive E-mail Messages	109
Object Query Listview Does Not Display Data	110
Operation Fails with Package not Found Error	110
Operation Fails with Login Failed Error	110
Sybase Settings Program Does Not Appear on Windows Mobile Device	110
Failed to Create Second Process	111
Customized Code is Blocked	111
Troubleshoot Device Clients	113
Device Provisioning	113
Device Cannot Communicate with Afaria Server	113
Device Clients	113
Synchronization Fails in a Remote Server Environment	113

Synchronization Fails With OutOfMemory Error	114
Timeout Errors During Client Synchronization ...	114
Device Operation Fails With	
SQLE_NOT_IMPLEMENTED Error	115
No Data Displays on Device Client	115
Synchronizing the Device Application Returns	
Only One Row of Data	116
Packet Dropped State	116
Pattern is Too Long Exception	118
Feature Not Implemented Exception	118
BlackBerry	119
iOS	121
Windows Mobile and Windows	123
Validation Rules	125
Mobile Business Object Validation Rules and Error	
Messages	125
Mobile Business Object, Attribute, and Structure	
Object Error Messages and Validation Rules	
.....	126
Parameter and Argument Validation Rules and	
Error Messages	132
Operation Validation Rules and Error Messages	
.....	138
Relationship Validation Rules and Error	
Messages	140
Cache Validation Rules and Error Messages ...	146
Synchronization Validation Rules and Error	
Messages	148
Object Query Validation Rules and Error	
Messages	150
Personalization Key Validation Rules and Error	
Messages	152

Custom Filters, Roles, and Synchronization Group Validation Rules and Error Messages	153
Mobile Workflow Validation Rules and Error Messages	154
Glossary: Sybase Unwired Platform	163
Index	175

Introduction to Troubleshooting

This troubleshooting guide provides information about resolving configuration, development, deployment, and production problems that you may encounter using Sybase® Unwired Platform.

The troubleshooting guide includes a collection of symptoms, with information to further diagnose or solve the problem. These presenting symptoms are organized into categories. You can either check the appropriate category for a problem, or you can search the *Troubleshooting* online help for the symptom you have experienced, or for an error message you have observed.

Troubleshooting categories include:

- Basic scenarios
- Sybase Unwired Platform system
- Sybase Control Center for Sybase Unwired Platform
- Mobile business object development
- Device application development
- Device clients
- Validation rules for mobile business objects processing

Use the troubleshooting guide with other Sybase Unwired Platform documents.

Documentation Roadmap for Unwired Platform

Learn more about Sybase® Unwired Platform documentation.

Table 1. Sybase Unwired Platform Documentation

Document	Description
<i>Sybase Unwired Platform Installation Guide</i>	<p>Describes how to install or upgrade Sybase Unwired Platform. Check the <i>Sybase Unwired Platform Release Bulletin</i> for additional information and corrections.</p> <p>Audience: IT installation team, training team, system administrators involved in planning, and any user installing the system.</p> <p>Use: during the planning and installation phase.</p>

Document	Description
<p><i>Sybase Unwired Platform Release Bulletin</i></p>	<p>Provides information about known issues, and updates. The document is updated periodically.</p> <p>Audience: IT installation team, training team, system administrators involved in planning, and any user who needs up-to-date information.</p> <p>Use: during the planning and installation phase, and throughout the product life cycle.</p>
<p><i>New Features</i></p>	<p>Describes new or updated features.</p> <p>Audience: all users.</p> <p>Use: any time to learn what is available.</p>
<p><i>Fundamentals</i></p>	<p>Describes basic mobility concepts and how Sybase Unwired Platform enables you to design mobility solutions.</p> <p>Audience: all users.</p> <p>Use: during the planning and installation phase, or any time for reference.</p>
<p><i>System Administration</i></p>	<p>Describes how to plan, configure, manage, and monitor Sybase Unwired Platform. Use with the <i>Sybase Control Center for Sybase Unwired Platform</i> online documentation.</p> <p>Audience: installation team, test team, system administrators responsible for managing and monitoring Sybase Unwired Platform, and for provisioning device clients.</p> <p>Use: during the installation phase, implementation phase, and for ongoing operation, maintenance, and administration of Sybase Unwired Platform.</p>

Document	Description
<i>Sybase Control Center for Sybase Unwired Platform</i>	<p>Describes how to use the Sybase Control Center administration console to configure, manage and monitor Sybase Unwired Platform. The online documentation is available when you launch the console (Start > Programs > Sybase > Sybase Control Center, and select the question mark symbol in the top right quadrant of the screen).</p> <p>Audience: system administrators responsible for managing and monitoring Sybase Unwired Platform, and system administrators responsible for provisioning device clients.</p> <p>Use: for ongoing operation, administration, and maintenance of the system.</p>
<i>Troubleshooting</i>	<p>Provides information for troubleshooting, solving, or reporting problems.</p> <p>Audience: IT staff responsible for keeping Sybase Unwired Platform running, developers, and system administrators.</p> <p>Use: during installation and implementation, development and deployment, and ongoing maintenance.</p>
Tutorials	<p>Tutorials for trying out basic development functionality.</p> <p>Audience: new developers, or any interested user.</p> <p>Use: after installation.</p> <ul style="list-style-type: none"> • Learn mobile business object (MBO) basics, and create a mobile device application: <ul style="list-style-type: none"> • <i>Tutorial: Mobile Business Object Development</i> • Create native mobile device applications: <ul style="list-style-type: none"> • <i>Tutorial: BlackBerry Application Development</i> • <i>Tutorial: iOS Application Development</i> • Create a mobile workflow package: <ul style="list-style-type: none"> • <i>Tutorial: Mobile Workflow Package Development</i>
<i>Sybase Unwired WorkSpace – Mobile Business Object Development</i>	<p>Online help for developing MBOs.</p> <p>Audience: new and experienced developers.</p> <p>Use: after system installation.</p>

Document	Description
<i>Sybase Unwired WorkSpace – Mobile Workflow Package Development</i>	<p>Online help for developing mobile workflow applications.</p> <p>Audience: new and experienced developers.</p> <p>Use: after system installation.</p>
Developer guides for device application customization	<p>Information for client-side custom coding using the Client Object API.</p> <p>Audience: experienced developers.</p> <p>Use: to custom code client-side applications.</p> <ul style="list-style-type: none"> • <i>Developer Guide for BlackBerry</i> • <i>Developer Guide for iOS</i> • <i>Developer Guide for Mobile Workflow Packages</i> • <i>Developer Guide for Windows and Windows Mobile</i>
Developer guide for Unwired Server side customization – <i>Developer Guide for Unwired Server</i>	<p>Information for custom coding using the Server API.</p> <p>Audience: experienced developers.</p> <p>Use: to customize and automate server-side implementations for device applications, and administration, such as data handling.</p> <p>Dependencies: Use with <i>Fundamentals</i> and <i>Sybase Unwired WorkSpace – Mobile Business Object Development</i>.</p>
Developer guide for system administration customization – <i>Developer Guide for Unwired Server Management API</i>	<p>Information for custom coding using administration APIs.</p> <p>Audience: experienced developers.</p> <p>Use: to customize and automate administration at a coding level.</p> <p>Dependencies: Use with <i>Fundamentals</i> and <i>System Administration</i>.</p>

Troubleshooting Basics

Sybase Unwired Platform provides tools for gathering information about system health properties. The Sybase Control Center administrative console, debugging, and tracing, as well as native operating system commands, and Unwired Platform command line tools combined offer root-cause diagnosis and problem resolution abilities to development and operational staff working with Sybase Unwired Platform.

The primary goals of troubleshooting:

- Gather enough information to diagnose the problem, so you can solve it.
- Gather supporting information to document the problem, so you can contact Sybase Technical Support for help in resolving it.

Collecting Information

Use Sybase Unwired Platform and other resources to collect information, then use the information to troubleshoot and solve the problem.

Sybase recommends that you develop a baseline for Unwired Platform system operations when things are running smoothly. This provides a benchmark for comparison when you encounter problems, and can lead to ideas for where to focus your troubleshooting efforts.

Unwired Platform resources include the Sybase Control Center Web console, error messages, log files, monitoring facilities, and the trace feature. Additional system resources include the system event log, operating system monitoring utilities, and third-party utilities.

The Visual Studio debugger is also a good resource for Windows and Windows Mobile device applications; use it to walk through code on the client user interface side.

Another resource is the Solved Cases database on the Sybase Support and Services Web site: *Solved Cases* (<http://www.sybase.com/support>).

Sybase Control Center Console Messages

Sybase Control Center for Sybase Unwired Platform can be used to review an ongoing stream of system messages to keep you alert to system health and processing status. System messages are captured in log files, and aggregated for display.

Use the Sybase Control Center administrative console to view system and error messages on an ongoing basis. See the *Sybase Control Center for Sybase Unwired Platform* online documentation for more information about using the console to view messages and log files.

Error Message Severity Levels

Error message severity levels provide an indication of system health, and how seriously to take a message.

Do not view individual messages in isolation, however, since a system event may trigger a range of symptoms that individually may not seem related. Review messages as a group, look for trends, and view upstream messages in the log for possible initiating events.

Table 2. Error message severity levels

Severity level	Description
Trace	Finer-grained information than Debug.
Debug	Fine-grained informational events that are useful for debugging an application.
Info	Informational messages that highlight application progress at a coarse-grained level.
Warn	Warning messages that indicate a potentially harmful situation.
Error	Error events that may allow the application to continue running.
Console (off)	Messages sent to the <code>system.out</code> stream logs for the Sybase Control Center administrative console.

Variables in Error Message Text

Unwired Platform error messages are frequently composed of one or more variables. Unwired Platform components, such as Unwired Server, and back-end enterprise information systems (EIS) replace the variable with the appropriate system information.

This informational message includes several variables:

```
INFO: SUCCESS:<Userid>:<MBO name>[.Operation name]:<effective parameter-value pairs>: [Warning message from EIS]
```

This warning message includes several variables:

```
WARN: FAILED:<Userid>:<MBO name>[.Operation name]:<effective parameter-value pairs>: <Error message from EIS>
```

Variables are shown in brackets:

- `<Userid>` – a user id will be inserted in the error message.
- `<MBO name>[.Operation name]:<effective parameter-value pairs>` – a string of variable information will be inserted, including the mobile business object name, an operation name, and parameter-value pairs.
- `<Warning/Error message from EIS>` – a warning or error message from the enterprise information system (EIS) will be inserted.

Log Files

Unwired Platform log files provide a source of troubleshooting information. Log file information is aggregated and displayed in the Sybase Control Center administrative console.

Accessing server logs

The server log records all system events as per the log level configuration for individual sub-components. For detailed diagnostics, you can configure individual sub-system's log level to DEBUG or TRACE level, and the messages recorded in the server log can help identify the problem. It is not recommended to leave sub-system log levels to those considering the performance impact and growth of the log file when large number of users may be accessing the system. Server log is separate for each server.

Server log settings are controlled from Sybase Control Center. Server log configuration if done on primary node will be propagated to other nodes at the next cluster changing action (package deployment).

In Sybase Control Center, navigate to *Servers > <ServerName> Log node > General tab*.

You can also view the server log from the command line using a text editor or any other third-party log viewing tool. (Sybase Control Center helps to view the server log remotely without needing access to the server's file system).

- The main server log file is `<UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\UnwiredServer\logs\<hostname>-server.log`.
- The incoming HTTP requests log (for Data Change Notification, and others) is in the `<hostname>-http.log` file. You may also see some additional folders which contain log files produced by the internal modules of the Messaging Server. The key folders are JMSBridge and SUPBridge. The messages recorded in those files are controlled by the log level setting of the MSG sub-component of the Unwired Server logging system.

These key sub-systems can help troubleshoot issues:

- Data Services – consider modifying this sub-system log level if you are trying to diagnose issues with backend EIS interaction, caching, and Data Change Notification.
- MMS – considering modifying this sub-system log level if you are trying to diagnose issues with client interaction, message-based synchronization application activities such as subscription command processing, data import messages, etc., and replication-based synchronization application activities such as those during upload and download phases of the synchronization, data change notification processing, background tasks such as cache refresh, etc. among others.
- MSG – consider modifying this sub-system if you are need to look at the individual messages being exchanged between one or more messaging-based synchronization application clients and server.

Accessing domain logs

The domain log shows data only if domain logging is enabled for that domain, and one of the packages in the domain has package-level logging enabled. Domain log data is automatically purged based on the domain log setting for the domain. Domain log is a cluster-wide logical view of the activities and errors.

Navigate to *Domains > <DomainName> > Log node > General tab*.

Accessing client logs

If client logging is enabled through the client object API, the client logs provide critical information about the application running on the device and its operational state. The client application information is uploaded from the device to Unwired Server, where it is visible using Sybase Control Center.

In the left navigation pane, expand the Packages folder and select the package you want to investigate, then in the right pane, select the **Client Log** tab.

Server Logs

Server logs provide a view of Unwired Server activity. Status and error messages are collected. You can set the level of granularity for messages captured, and the log file size through Sybase Control Center. In a troubleshooting situation, you may want to increase the granularity of messages captured.

Search *Sybase Control Center for Sybase Unwired Platform* online help for the topic *Configuring Server Log Settings* for information.

Domain Logs

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

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

Client Logs

Client logs capture data about client application operations for all devices subscribed to a package. You can use the data to track errors and identify performance issues.

By default, logged data includes the user name, device ID, mobile business object with which the client is syncing, the operation the client is performing, the success or failure of the server-side operation, current log level, date and time stamp, the log message associated with the message.

Developers use client-side APIs to enable this capability to log critical application data on the device that can be uploaded automatically to the server where it is visible in the package's client log. The capability can be implemented so that enabling and disabling logging does not require any user intervention, is transparent, and can be turned on or off automatically.

See *Sybase Control Center > Manage > Managing Unwired Platform > Routine System Maintenance Tasks > Checking Client Application Logs*.

Log File Locations

Use a text editor to review log files from the command line using a text editor if Sybase Control Center is not available, or to concentrate your review to a particular log.

Table 3. Log file locations

Log type	Location
Unwired Server	<p>Aggregated Unwired Server logs, including replication-based synchronization events and SYSAM issues: <code><UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\UnwiredServer\logs</code></p> <p>Messaging service log details: <code><UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\UnwiredServer\logs\<Module></code></p> <p>Device client tracing logs: <code><UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\MessagingServer\Data\ClientTrace</code></p> <p>Mobile Workflow tracing logs: <code><UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\UnwiredServer\logs\WorkflowClient</code></p>
Sybase Control Center for Sybase Unwired Platform	<p>Sybase Control Center agent log: <code><UnwiredPlatform_InstallDir>\SCC-XX\log\agent.log</code></p> <p>Repository log: <code><UnwiredPlatform_InstallDir>\SCC-XX\log\repository.log</code></p> <p>Request logs: <code><UnwiredPlatform_InstallDir>\SCC-XX\services\EmbeddedWebContainer\log\request-<yyyy_mm_dd>.log</code></p> <p>Database message: <code><UnwiredPlatform_InstallDir>\SCC-XX\services\Repository\scc_repository.slg</code></p>

Log type	Location
Relay server and RSOE	<p>Default log details:</p> <p><code>%temp%\ias_relay_server_host.log</code></p> <p><code>%temp%</code> is the Windows environment variable, for example, <code>C:\WINDOWS\Temp</code>.</p> <hr/> <p>Note: In the case of IIS Web servers, the log file is <code>C:\WINDOWS\system32\LogFiles</code>. However, this location can be configurable in IIS Manager.</p> <hr/> <p>RSOE log files, by default:</p> <p><code><UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\UnwiredServer\logs</code></p> <p>These log files are generated in the <code>\logs</code> directory for the corresponding RSOE:</p> <ul style="list-style-type: none"> • <code>rsoe.log</code> • <code>msgrsoe.log</code> • <code>webserver_rsoe.log</code>
Installer	<p><code><UnwiredPlatform_InstallDir></code></p> <p><code><UnwiredPlatform_InstallDir>\UnwiredPlatform\InstallLogs</code></p> <p><code><UnwiredPlatform_InstallDir>\UnwiredPlatform\InstallLogs\silentInstall</code></p>
Domain	Added to the monitoring database; viewable in Sybase Control Center
Consolidated database	By default, consolidated database errors are logged in the <code>error-log.txt</code> file, located in <code><UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\UnwiredServer</code> .

Reporting Errors

Before reporting problems to Sybase Technical Support, collect as much relevant background information as you can.

For application related issues, the relevant information would be the original Eclipse project that contains the mobile business objects and the client project, server log files, error messages, symptoms, observations, and steps you have taken to solve the problem.

For server administration issues, relevant information would be in the server log files and Sybase Control Center log files.

This information may help the technician reproduce and diagnose the problem.

Reproducing Problems

A problem that can be reproduced may help your technician more quickly solve the problem. "Reproducible" means that the code causing the incorrect behavior can be isolated, and that the issue can be demonstrated. It may also be useful to narrow the reproduction by removing extraneous steps or code.

Provide as much useful information as possible on your problem report, including a reproducible case.

Sybase Technical Support Checklist

Before reporting errors, collect information about Sybase Unwired Platform components.

Component	Information
Unwired Server, clients, applications, devices, users	<ul style="list-style-type: none"> • Unwired Server version and build; the version information is available in the server log as well displayed in the Sybase Control Center. • Full description of incorrect behavior, including all components involved • Server and client errors (stream error) and error messages (Unwired Server log files) • Client device platform, version, and build (if appropriate) • Application information (if appropriate) • User name and identifier information • Consolidated database error log file, which by default is <code>errorlog.txt</code> in <code><UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\UnwiredServer</code>. • For administration console issues, include <code>agent.log</code> and <code>repository.log</code> files.
Relay server, domains, clusters [separate installation]	<ul style="list-style-type: none"> • Relay Server version and build • Full description of incorrect behavior, including all components involved • Relay Server-related errors and error messages • Type of Web server (IIS or Apache), version, and operating system
Afaria [separate installation]	<ul style="list-style-type: none"> • Afaria® version and build • Full description of incorrect behavior, including all components involved • Afaria-related errors and error messages

Sending Error Log Fragments

Error logs are a valuable source of information for troubleshooting problems. Provide error logs, or error log fragments, when reporting your problem. Also provide a timestamp of when the problem occurred, so that logs can be compared.

In many cases it is advisable to back up the log file, clear its contents, reproduce the problem, and then send log file. This reduces the content of the log file to pertinent information needed to diagnose the problem.

If possible, remove or un-install un-necessary packages from Unwired Server, then reproduce the problem. When reproducing the problem, try, as much as possible, to omit unnecessary steps. If you have cleared the log, as described above, before reproducing the problem, only relevant information will be logged. Another good idea for capturing diagnostic information is to take error screen shots, or record videos that capture reproducible steps. These practices significantly reduce unnecessary information in server logs and help the troubleshooting process.

To prepare error log file fragments:

1. Identify the log files that may include important information.
2. Copy error log files to a separate directory, or copy a section from an error log file to a separate text file.
3. Include the error log file or files, as well as screen shots and video logs, with the problem report you submit to Technical Support.

Ensuring System Recoverability

Ensure system recoverability by keeping Sybase Unwired Platform components in good health, understanding your topology, and implementing sound operations and maintenance practices. Develop a disaster recovery plan that includes backing up the installation directory, consolidated database, and transaction logs.

See *System Administration* for information about maintaining system health, and backing up and restoring Sybase Unwired Platform.

Troubleshoot Basic Scenarios

Troubleshooting information for debugging process flow for the system, and for device application development; identifies useful tools and utilities.

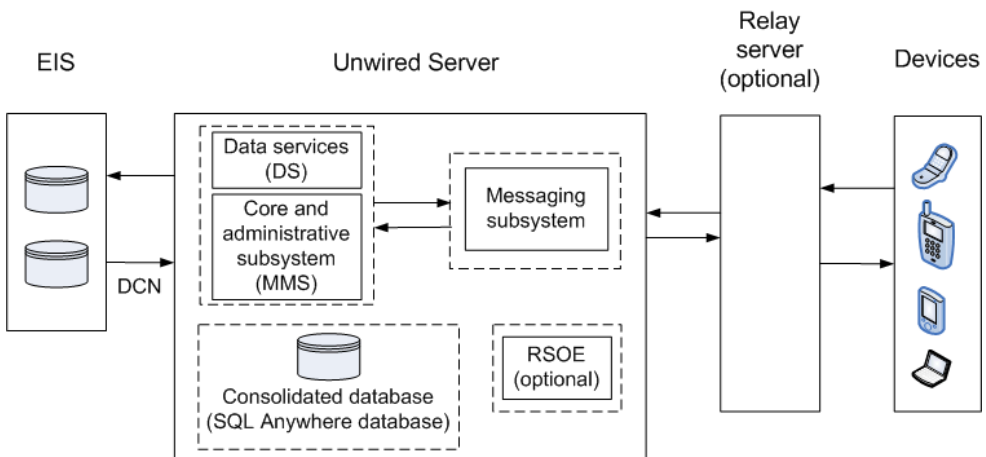
Sybase Unwired Platform End-to-End Process Flow

Understanding Sybase Unwired Platform processing flow at a high level between the enterprise information system (EIS), Sybase Unwired Platform, and devices is important for troubleshooting, as well as knowing the options available for diagnosing problems at each step of the process.

High-level process flow

This diagram shows a logical view of Sybase Unwired Platform communications between the EIS, Sybase Unwired Platform components, and device clients. The diagram also shows relay server and relay server outbound enabler (RSOE), which are optional components.

Figure 1: Sybase Unwired Platform communications



Unwired Server subcomponents include:

- Core and administrative subsystem (MMS) – responsible for core services which include device client application interaction management, security enforcement, notifications processing, and runtime component management. MMS administrative services provide the server management interface used by Sybase Control Center.
- Data services (DS) – responsible for back-end interaction, data integrity, and transactions.

- Consolidated database (CDB) – is a runtime cache database used by Unwired Server. By default, an embedded SQL Anywhere® database server is used as the CDB, as shown in the diagram. However, you can configure Unwired Server to use an existing SQL Anywhere instance as its CDB server.
- Messaging subsystem – responsible for transport of messages required for maintaining data integrity on device client applications. It uses a JMS service to upload and download data changes between devices and the consolidated database.

Troubleshooting options

This section describes the options available for diagnosing process flow problems. Details are provided separately.

Table 4. Options for diagnosing process flow

Processing component	Diagnostic option
Enterprise Information System (EIS)	Use EIS administrative tools to make sure the data source is available, and check logs for possible communications problems between the EIS and Unwired Server.
Unwired Server – data services (DS)	<ul style="list-style-type: none"> • Using Sybase Control Center, enable data services subsystem logging. • Review interactions for getting data and sending transactions.
Unwired Server – core and administrative subsystem (MMS)	<ul style="list-style-type: none"> • Using Sybase Control Center, enable core and administrative subsystem logging. • Monitor packages. • Enable package-level logging. • Enable synchronization tracing. • View the mobile business object (MBO) operation history.
Unwired Server – messaging subsystem	<ul style="list-style-type: none"> • Using Sybase Control Center, enable messaging subsystem logging. • Enable messaging server-side logging for one or more of its modules. • Retrieve client-side messaging-based synchronization trace by sending a request to the messaging device from Sybase Control Center. • Check the module log for SUPBridge and JMSBridge modules messages. The SUPBridge log contains incoming messages from device clients, and the JMSBridge log contains messages going to device clients.

Processing component	Diagnostic option
Relay server (RSOE) – if installed	<ul style="list-style-type: none"> • Enable relay server tracing. • Review relay server log. • Check Microsoft Windows system log files. • Check relay server configuration.
Device clients	<ul style="list-style-type: none"> • If the SUPLogger is implemented as part of custom coding, enable Show Log, and set the trace flags. • Enable WorkflowClient tracing from the client. • Use Sybase Control Center to check device information and settings, including the current connection status. • Review log files uploaded from the device. • Check the configuration, especially connections between the device and Unwired Server, and the certificate.

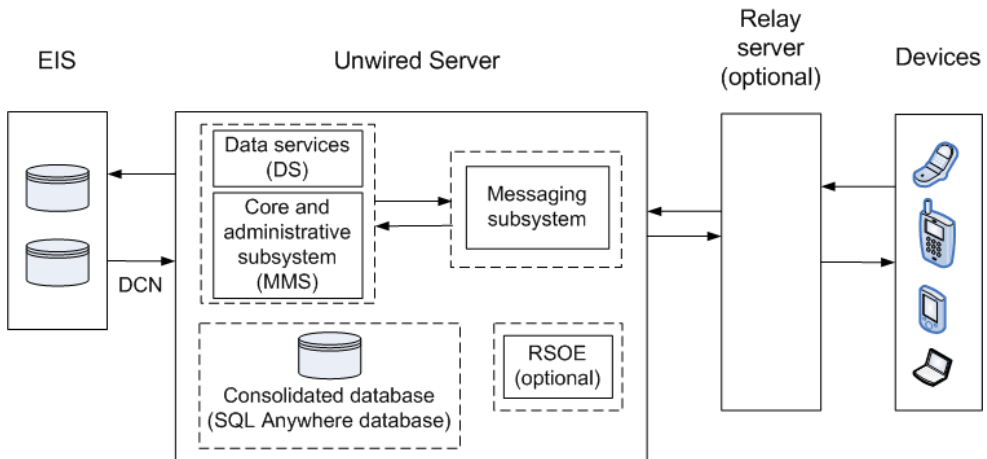
Debugging Message-Based Synchronization

Describes the message-based synchronization process flow, and options available for diagnosing problems at each step of the process.

Process flow

This diagram shows a logical view of message-based synchronization communications between the enterprise information system (EIS), Sybase Unwired Platform components, and device clients. Relay server is included as an optional component.

Figure 2: MBS process flow



Troubleshooting options

This section describes the options available for diagnosing MBS process flow problems.

Unwired Server – data services (DS):

- In Sybase Control Center, enable data services (DS) subsystem logging by setting its log level. The output is written in the server log.
- Review subsystem messages for information about getting data and sending transactions.
- For cache activities, you can enable monitoring on the package and then use Sybase Control Center to review cache statistics captured for the period monitoring is enabled.

Unwired Server – core and administrative subsystem (MMS):

- If you like to troubleshoot at the system-level (which will record all domains and packages activities), you can enable MMS logging by setting the log level in Sybase Control Center. The output is written in the server log.
- Select the desired application package, and enable monitoring. The captured data appears in Sybase Control Center. Use package-level logging to review data in the domain log.
- For low-level debugging and diagnostics, you may enable "Sync Tracing" for the package in Sybase Control Center. The output is written in the server log.
- For quick troubleshooting to locate system errors, you can check the package-level client log and MBO/Operation history in the Sybase Control Center.
- View the MBO/Operation history in Sybase Control Center for errors during back-end interaction with the EIS.
- On Unwired Server, review the `errorlog.txt` for any consolidated database system errors: `<UnwiredPlatform_InstallDir\UnwiredPlatform\Servers\UnwiredServer\errorlog.txt`.

Unwired Server – messaging subsystem :

- Use Sybase Control Center to enable server-side logging for <MSG> log subcomponents.
- Retrieve client-side message-based synchronization trace by sending a request to the MBS device using Sybase Control Center (on the **Devices** tab, use **Get trace**). A client's log level can be set in the device settings of the Sybase Control Center.
- Configure messaging server-side module level tracing (for users with advanced knowledge).
 - The message service uses `TraceConfig.xml` where log levels of individual modules can be set.
 - If debugging a Mobile Workflow application, turn on `WorkflowClient` tracing from the client.
 - Check the module logs – `SUPBridge` and `JMSBridge` modules, or the `WorkflowClient` module (to debug Mobile Workflow applications). In Sybase Control Center, the server log system merges logs from these modules.
- On Unwired Server, review the `errorlog.txt` for any consolidated database system errors: `<UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\UnwiredServer\errorlog.txt`.

Relay server – if used:

- Enable relay server tracing by setting verbosity to 1 or higher. You can do that by editing the `rs.config` file of your Relay Server and running the **rshost** utility. Check the *System Administration* topic on Relay Server Setup.
- Review the relay server log files. The logging for `rshost.exe` is directed to the `myrshost.log` file in the `C:\temp` directory.
- Check the IIS log files (`windows\logfiles\W3SVC1` and `HTTPERR`). The IIS server log path is configurable from `inetmgr` from **Default Web Site > Properties > Website tab > Enable Logging > Properties > Log file directory**. The recommended location is `C:\Inetpub\LogFiles`.

On Unwired Server, check the SUP outbound enabler log for messaging based synchronization entries: `<UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\UnwiredServer\Logs\<SUPServerName>.MBS<PORT>RSOE<num>.log` (for example, `supqa01SUPServer1a.MBS5001RSOE2.log`).

Device clients:

- For Windows and Windows Mobile, run the **Sybase Settings** application, and choose **Show Log**. The client system log for messaging activities appears. Key information includes messages related to server connectivity, server messages, and messaging client requests.

- Use the platform-specific client logging API (**Sybase.Persistence.ILogRecord** in C#, and **SUPLogger** in Objective-C) to capture log messages and upload them (Client Log) to Unwired Server. You can view these messages in the Sybase Control Center user interface.
- Set the device log trace level on the device and get the device logs from Sybase Control Center by selecting **Device Users > <device> > Properties > Device Advanced >** and setting the **Debug Trace Level** to 4 or higher.

After running the application on a device, select the device in Sybase Control Center and select **Get Trace**. The client trace log on the device is transferred to the Unwired Server at `<UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\MessagingServer\Data\ClientTrace\`.

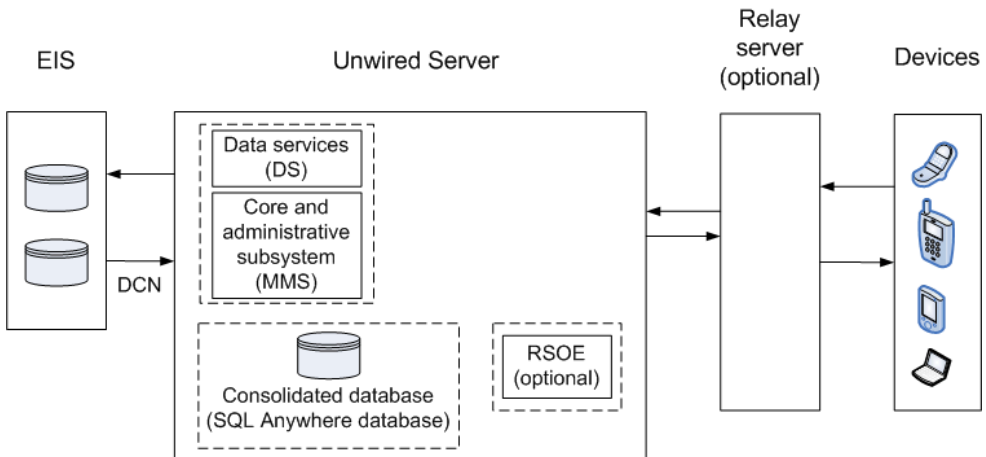
Debugging Replication-Based Synchronization

Describes the replication-based synchronization (RBS) process flow, and options available for diagnosing problems at each step of the process.

Process flow

This diagram shows a logical view of replication-based synchronization communications between the enterprise information system (EIS), Sybase Unwired Platform components, and device clients. Relay server is included as an optional component.

Figure 3: RBS process flow



Troubleshooting options

This section describes the options available for diagnosing RBS processing flow problems.

Unwired Server – data services (DS):

- In Sybase Control Center, enable data services (DS) subsystem logging by setting its log level. The output is written in the server log.
- Review subsystem messages for information about getting data and sending transactions.
- For cache activities, you can enable monitoring on the package and then use Sybase Control Center to review cache statistics captured for the period monitoring is enabled.

Unwired Server – core and administrative subsystem (MMS):

- If you like to troubleshoot at the system-level (which will record all domains and packages activities), you can enable MMS logging by setting the log level in Sybase Control Center. The output is written in the server log.
- Select the desired application package, and enable monitoring. The captured data appears in Sybase Control Center. Use package-level logging to review data in the domain log.
- For low-level debugging and diagnostics, you may enable "Sync Tracing" for the package in Sybase Control Center. The output is written in the server log.
- For quick troubleshooting to locate system errors, you can check the package-level client log and MBO/Operation history in the Sybase Control Center.
- View the MBO/Operation history in Sybase Control Center for errors during back-end interaction with the EIS.
- On Unwired Server, review the `errorlog.txt` for any consolidated database system errors: `<UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\UnwiredServer\errorlog.txt`.

Relay server – if used:

- Enable relay server tracing by setting verbosity to 1 or higher. You can do that by editing the `rs.config` file of your Relay Server and running the `rshost` utility. Check the *System Administration* topic on Relay Server Setup.
- Review the relay server log files. The logging for `rshost.exe` is directed to the `myrshost.log` file in the `C:\temp` directory.
- Check the IIS log files (`windows\logfiles\W3SVC1` and `HTTPERR`). The IIS server log path is configurable from `inetmgr` from **Default Web Site > Properties > Website tab > Enable Logging > Properties > Log file directory**. The recommended location is `C:\Inetpub\LogFiles`.

On Unwired Server, check the SUP outbound enabler log for replication based synchronization entries: `<UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\UnwiredServer\Logs\<SUPServerName>.RBS<PORT>RSOE<num>.log` (for example, `supqa01SUPServer1a.RBS2480RSOE1.log`).

Device clients:

- Use the platform-specific client logging API (**com.sybase.persistence.LogRecord** in Java, and **Sybase.Persistence.ILogRecord** in C#) to capture log messages and upload

them (Client Log) to Unwired Server. You can view these messages in the Sybase Control Center user interface

Debugging the Cache

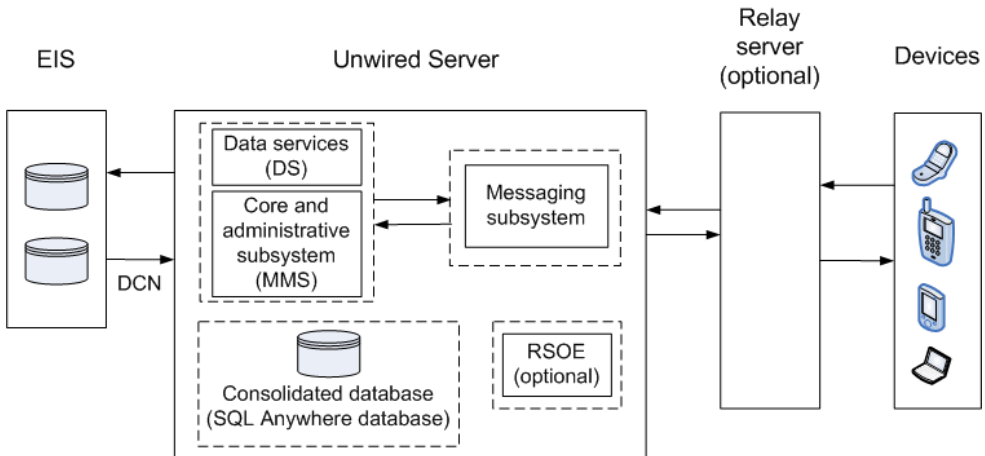
Describes the consolidated database cache process flow, and options available for diagnosing problems at each step of the process.

Process flow

This diagram shows a logical view of the cache process flow on Sybase Unwired Platform.

The cache refresh process is either scheduled (repeated at certain intervals) or on-demand when the client initiates a request. The cache can also be updated by incoming Data Change Notification (DCN) requests. A mobile business object (MBO) operation marked as `Invalidate the cache` operation forces cache contents to be refreshed.

Figure 4: Cache process flow



Troubleshooting options

This section describes the options available for diagnosing consolidated database cache processing flow problems.

Unwired Server – data services (DS):

- In Sybase Control Center, enable data services (DS) subsystem logging by setting its log level. The output is written in the server log.
- Review subsystem messages for information about getting data and sending transactions.
- For cache activities, you can enable monitoring on the package and then use Sybase Control Center to review cache statistics captured for the period monitoring is enabled.

In Sybase Control Center, check cache group properties such as schedule, and cache interval. Cache updates are propagated to the client based on a combination of settings for Synchronization group change detection interval, push related device settings (MBS), and notification threshold for subscriptions (RBS). You can view all those properties in the Sybase Control Center .

You can also use monitoring to trace the cache refresh timestamp, notification generation timestamp, data import message generation (MBS), or sync request from the client for local database update. See *System Administration* for further details on cache.

Unwired Server – Core and administrative subsystem (MMS):

- If you like to troubleshoot at the system-level (which will record all domains and packages activities), you can enable MMS logging by setting the log level in Sybase Control Center. The output is written in the server log.
- Select the desired application package, and enable monitoring. The captured data appears in Sybase Control Center. Use package-level logging to review data in the domain log.
- For low-level debugging and diagnostics, you may enable "Sync Tracing" for the package in Sybase Control Center. The output is written in the server log.
- For quick troubleshooting to locate system errors, you can check the package-level client log and MBO/Operation history in the Sybase Control Center.
- View the MBO/Operation history in Sybase Control Center for errors during back-end interaction with the EIS.
- On Unwired Server, review the `errorlog.txt` for any consolidated database system errors: `<UnwiredPlatform_InstallDir\UnwiredPlatform\Servers\UnwiredServer\errorlog.txt`.

Debugging Data Change Notification

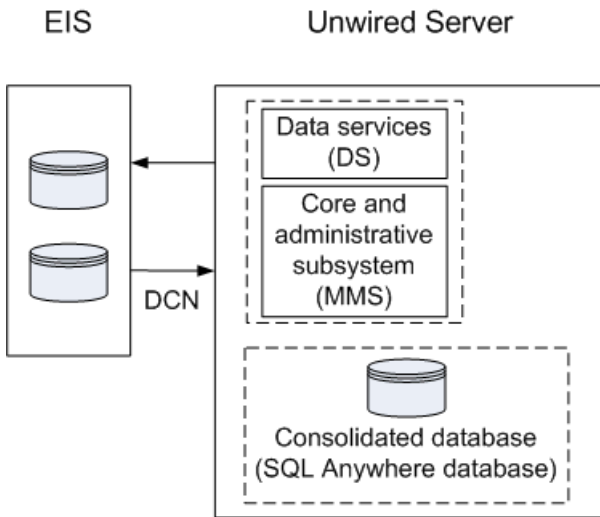
Describes the data change notification process flow, and options available for diagnosing problems at each step of the process.

Process flow

This diagram shows a logical view of data change notification (DCN) between the enterprise information system (EIS) and Sybase Unwired Platform components.

DCNs notify Unwired Server when data used by an application changes on an EIS server. DCNs are used as a strategy to update very specific data automatically. See *Reference: Administration APIs* and *System Administration* for development and administration information. See also *Reference: Custom Development for Unwired Server* for the DCN format and other details.

Figure 5: DCN process flow



DCN Error Reporting

Errors while processing data change notification (DCN) requests are reported with HTTP error codes. A single DCN request can contain multiple DCN messages and each message can have payload for a specific mobile business object (MBO).

If one of the DCN messages in a DCN request fail, then all the other successfully executed DCN messages are rolled back and a failure is reported as a DCN result with individual status messages for each DCN message. DCN messages that occur after the failed message are not processed, and failure is reported for them as well.

Success is reported only if all the DCN messages succeed.

Table 5. Error codes applicable to DCN

Error code	Error reason	Example of error cause
400	DCN usage error	A required parameter is missing from the DCN request.
401	Authentication error	The user name or password is incorrect.
403	Authorization error	The user does not have the SUPDCNUser assigned.
500	Internal server error	The DCN filter could not be loaded.

Table 6. DCN usage errors, HTTP error 400

Error	Error message
JSON parsing error	DCN JSON request <i><jsonDCNRequest></i> could not be parsed. Make sure it conforms to DCN JSON standard.
Required parameters not specified in DCN request like operation name, package name, user-name, password, dcn_request	<i><parameter></i> not supplied in the DCN request.
Wrong MBO name in DCN request	VirtualTableName is null. MBO name <i><mbo-Name></i> might be incorrect or missing capitalization in the DCN request.

Table 7. Sybase Unwired Platform internal errors, HTTP error code 500

Error code	Error reason
Wrong attributes in DCN request	Unable to set attribute names to update the cache. Check the cols section in the DCN request.
DS error in modifying cache	As returned by the DS.
DCN filter classloading issues	Package <i><packageName></i> or Server wide loader for filter class <i><filterClassName></i> not found. Check if DCN request contains the fully qualified name of the filter class and/or it is deployed on the SUP server.
DCN filter processing errors	Error in postprocessing/preprocessing by filter class <i><classname></i> .
Error in setting domain	Error in setting the current domain
Package properties not found in the SUP repository	Package properties not found for package <i><packageName></i> in domain <i><domainName></i> .
Error in getting DS object to process DCN	dsAPIServiceNotFound.
Error dispatching cache event by DS	As returned by the DS.

Troubleshooting options

This section describes the options available for diagnosing DCN processing flow problems.

EIS

- Verify the EIS data source (third party application, database, Web service) is running.
- Verify the DCN code is operating correctly.

Similar to cache, updates from DCN activity are propagated to devices based on a number of other settings. Review the change detection interval property associated with the synchronization group, the notification threshold of RBS subscription, or the push-related settings for the device (MBS).

Unwired Server – data Services (DS):

- In Sybase Control Center, enable data services (DS) subsystem logging by setting its log level. The output is written in the server log.
- Review subsystem messages for information about getting data and sending transactions.
- For cache activities, you can enable monitoring on the package and then use Sybase Control Center to review cache statistics captured for the period monitoring is enabled.

Unwired Server – administrative Services (MMS):

- If you like to troubleshoot at the system-level (which will record all domain and package activities), you can enable MMS logging by setting the log level in Sybase Control Center. The output is written in the server log.
- Select the desired application package, and enable monitoring. The captured data appears in Sybase Control Center. Use package-level logging to review data in the domain log.
- On Unwired Server, review the `errorlog.txt` for any consolidated database system errors: `<UnwiredPlatform_InstallDir\UnwiredPlatform\Servers\UnwiredServer\errorlog.txt`.
- Enable monitoring to capture DCN request history and performance data.
- Check the DCN request log in the Unwired Server HTTP Request Log (`<UnwiredPlatform_InstallDir\UnwiredPlatform\Servers\UnwiredServer\logs\<hostname>-http.log`). The log shows a history of DCN requests, and performance data (including counts of total requests and processing time).

Debugging BlackBerry Device Development

Device client and Unwired Server troubleshooting tools for diagnosing RIM® BlackBerry® development problems.

Client-side debugging

Problems on the device client side that may cause client application problems:

- Unwired Server connection failed.
- Data does not appear on the client device.
- Physical device problems, such as low battery or low memory.

To find out more information on the device client side:

- If you have implemented debugging in your generated or custom code (which Sybase recommends), turn on debugging, and review the debugging information. See *Developer*

Guide for BlackBerry about using the `MBOLogger` class to add log levels to messages reported to the console.

- Check the log record on the device. Use the `<PkgName>DB.getLogRecords` (**com.sybase.persistence.Query**) or **Entity.getLogRecords()** methods. Use this method for logs corresponding to MBO classes.

This is the log format:

```
level,code,eisCode,message,component,entityKey,operation,requestId,timestamp
```

This is a log sample:

```
5,500,'','java.lang.SecurityException:Authorization failed:
Domain = default Package = end2end.rdb:1.0 mboName =
simpleCustomer action =
delete','simpleCustomer','100001','delete','100014','2010-05-11
14:45:59.710'
```

- `level` – the log level currently set. Values include: 1 = TRACE, 2 = DEBUG, 3 = INFO, 4 = WARN, 5 = ERROR, 6 = FATAL, 7 = OFF.
- `code` – replication-based synchronization, Unwired Server administration codes:
 - 200 – success.
 - 500 – failure.
- `eisCode` – not currently used.
- `message` – the message content.
- `component` – Mobile Business Object (MBO) name.
- `entityKey` – MBO surrogate key, used to identify and track MBO instances and data.
- `operation` – operation name.
- `requestId` – operation replay request ID or messaging-based synchronization message request ID.
- `timestamp` – message logged time, or operation execution time.
- Check the Storm event log:
 1. On the Home screen, press Hold.
 2. Click the upper-left corner and upper-right corner twice.
 3. Review the event log.
- Check the BlackBerry event log:
 1. On the device, press ALT+Iglg; or, for touch-screen devices, hold the ESC key, tap (no click) top-left, top-right, top-left, then top-right.
 2. Review the event log, and see the RIM BlackBerry documentation for information about debugging and optimizing. http://na.blackberry.com/eng/developers/resources/A50_How_to_Debug_and_Optimize_V2.pdf

Server-side debugging

Problems on the Unwired Server side that may cause device client problems:

- The domain or package does not exist.
- Authentication failed for the synchronizing user.
- The operation role check failed for the synchronizing user.
- Back-end authentication failed.
- An operation failed on the remote, replication database back end, for example, a table or foreign key does not exist. Detailed messages can be found in the Log Record.
- An operation failed on the Web service, REST, or SAP® back end. You can find detailed messages in the log record.

To find out more information on the Unwired Server side:

- Check the MMS server log files. See the *Sybase Control Center* documentation for more information.

Debugging iOS Device Development

Device client and Unwired Server troubleshooting tools for diagnosing Apple iOS development problems.

Client-Side Debugging

Problems on the device client side that may cause client application problems:

- Unwired Server connection failed.
- Data does not appear on the client device.
- Physical device problems, such as low battery or low memory.

To find out more information on the device client side:

- If you have implemented debugging in your generated or custom code (which Sybase recommends), turn on debugging, and review the debugging information. See *Developer Guide for iOS* about using the `MBOLogger` class to add log levels to messages reported to the console.
- Check the log record on the device. Use the individual entity class `getLogRecords` method, or assemble a query using the `SUPQuery` class and call the database class `getLogRecords` method.

This is a sample log format (depending on your query):

```
<header> [level]
           code
           component
           entityKey
           level
           message
           operation
           requestId
           timestamp
```

This log format generates output similar to:

```
2010-06-22 10:54:28.927 SAPSimpleApps2[91202:207] [INFO]
[SAPTest.m:295] entity has a log record:
    code = 200,
    component = SalesOrder,
    entityKey = 500151,
    level = 3,
    message = Successful,
    operation = changeFromData,
    requestId = 33416006,
    timestamp = 2010-06-22 03:54:22 -0700
```

- code – log record code numbers sent by Unwired Server, based on HTTP return codes.
 - Replication-based synchronization codes:
 - 200 – success.
 - 500 – failure.
 - Message-based synchronization codes:
 - 200 – success.
 - 401 – the client request included invalid credentials, or authentication failed for some other reason.
 - 403 – the client request included valid credentials, but the user does not have permission to access the requested resource (package, mobile business object —MBO, or operation).
 - 404 – the client tried to access a nonexistent package or MBO.
 - 405 – there is no valid license to check out for the client.
 - 409 – back-end EIS is deadlocked.
 - 412 – back-end EIS threw a constraint exception.
 - 500 – an unexpected (unspecified) server failure.
 - 503 – back-end EIS is not responding or the connection is terminated.
- eisCode – maps to HTTP error codes. If no mapping exists, defaults to error code 500 (an unexpected server failure).
- entityKey – if this log record refers to a particular row in an MBO table, this field contains a string representation of the primary key for the row.
- component – if this log record refers to a particular row in an MBO table, this field contains the name of the MBO.
- level – the log level currently set. Values include: 1 = Trace, 2 = Debug, 3 = Info, 4 = Warn, 5 = Error, and 6 = Fatal.
- message – the reason for the log record, for example, an error from the server when attempting to process a client replay request for a create, update, and delete operation.
- messageID – unique log record ID.
- operation – for log records written by Unwired Server, the operation (such as create, update, and delete) that caused this log record to be written.

- `requestId` – for log records written by Unwired Server, the ID of the replay request that caused this log record to be written.
- `timestamp` – the date and time the log record was created.
- If you have implemented `onConnectionStatusChange` for message-based synchronization in `CallbackHandler`, the connection status between Unwired Server and the device is reported on the device. See the *Developer Guide for iOS for CallbackHandler* information. The status reported on the device includes:
 - 0 – device connected.
 - 1 – device not connected.
 - 2 – device not connected because of flight mode.
 - 3 – device not connected because no network coverage.
 - 4 – device not connected and waiting to retry a connection.
 - 5 – device not connected because roaming was set to false and device is roaming.
 - 6 – device not connected because of low space.

Server-Side Debugging

Problems on the Unwired Server side that may cause device client problems:

- The domain or package does not exist.
- Authentication failed for the synchronizing user.
- The operation role check failed for the synchronizing user.
- Back-end authentication failed.
- An operation failed on the remote, replication database back end, for example, a table or foreign key does not exist.
- An operation failed on the Web Service, REST, or SAP back end. You can find detailed messages in the log record.

To find out more information on the Unwired Server side:

- Check the Unwired Server log files.
- For message-based synchronization mode, check the `SUPBridge` and `JMSBridge` logs.

Debugging Windows and Windows Mobile Device Development

Device client and Unwired Server troubleshooting tools for diagnosing Microsoft Windows and Windows Mobile development problems.

Client-Side Debugging

Problems on the device client side that may cause client application problems:

- Unwired Server connection failed.
- Data does not appear on the client device.
- Physical device problems, such as low battery or low memory.

To find out more information on the device client side:

- If you have implemented debugging in your generated or custom code (which Sybase recommends), turn on debugging and review the debugging information. See *Developer Guide for Windows and Windows Mobile* about using the `MBOLogger` class to add log levels to messages reported to the console.
- Check the log record on the device. Use the `DatabaseClass.GetLogRecord(Sybase.Persistence.Query)` or `Entity.GetLogRecords` methods.

This is the log format

```
level,code,eisCode,message,component,entityKey,operation,requestId,timestamp
```

This log format generates output similar to:

```
level code eisCode message component entityKey operation requestId
timestamp
 5,500,'','java.lang.SecurityException:Authorization failed:
Domain = default Package = end2end.rdb:1.0 mboName =
simpleCustomer action =
delete','simpleCustomer','100001','delete','100014','2010-05-11
14:45:59.710'
```

- `level` – the log level currently set. Values include: 1 = TRACE, 2 = DEBUG, 3 = INFO, 4 = WARN, 5 = ERROR, 6 = FATAL, 7 = OFF.
- `code` – Unwired Server administration codes.
 - Replication-based synchronization codes:
 - 200 – success.
 - 500 – failure.
- `eisCode` – maps to HTTP error codes. If no mapping exists, defaults to error code 500 (an unexpected server failure).
- `message` – the message content.
- `component` – MBO name.
- `entityKey` – MBO surrogate key, used to identify and track MBO instances and data.
- `operation` – operation name.
- `requestId` – operation replay request ID or messaging-based synchronization message request ID.
- `timestamp` – message logged time, or operation execution time.
- If you have implemented `onConnectionStatusChange` for message-based synchronization in `CallbackHandler`, the connection status between Unwired Server

and the device is reported on the device. See the *Developer Guide for Windows and Windows Mobile* for `CallbackHandler` information. The device connection status, device connection type, and connection error message are reported on the device:

- 1 – current device connection status.
- 2 – current device connection type.
- 3 – connection error message.

Server-Side Debugging

Problems on the Unwired Server side that may cause device client problems:

- The domain or package does not exist. If you create a new domain, whose default status is disabled, it is unavailable until enabled.
- Authentication failed for the synchronizing user.
- The operation role check failed for the synchronizing user.
- Back-end authentication failed.
- An operation failed on the remote, replication database back end, for example, a table or foreign key does not exist.
- An operation failed on the Web Service, REST, or SAP® back end.

To find out more information on the Unwired Server side:

- Check the Unwired Server log files.
- For message-based synchronization mode, you can set the log level to DEBUG to obtain detailed information in the log files:
 1. Check the global SUP MSG log level in `<server_install_folder>\UnwiredPlatform\Servers\UnwiredServer\Repository\logging-configuration.xml` to ensure the Log level of `<Entity EntityTypeId="MSG">` is set to DEBUG.
 2. Modify the log level for the module SUPBridge and JmsBridge in `<server_install_folder>\UnwiredPlatform\Servers\MessagingServer\Data\TraceConfig.xml` to DEBUG.
 3. Check the SUPBridge and JMSBridge logs, for detailed information.

Note: It is important to return to INFO mode as soon as possible, since DEBUG mode can effect system performance.

- You can also obtain DEBUG information for a specific device:
 - View information through the SCC administration console:
 1. Set the DEBUG level to a higher value for a specified device through SCC administration console:
 - a. On SCC, select a device, then select **Properties... > Device Advanced**.
 - b. Set the Debug Trace Level value.
 2. Set the TRACE file size to be more than 50KB.
 3. View the trace file through SCC.

- Check the `<server_install_folder>\UnwiredPlatform\Servers\MessagingServer\Data\ClientTrace` directory to see the mobile device client log files for information about a specific device.

Note: It is important to return to INFO mode as soon as possible, since DEBUG mode can effect system performance.

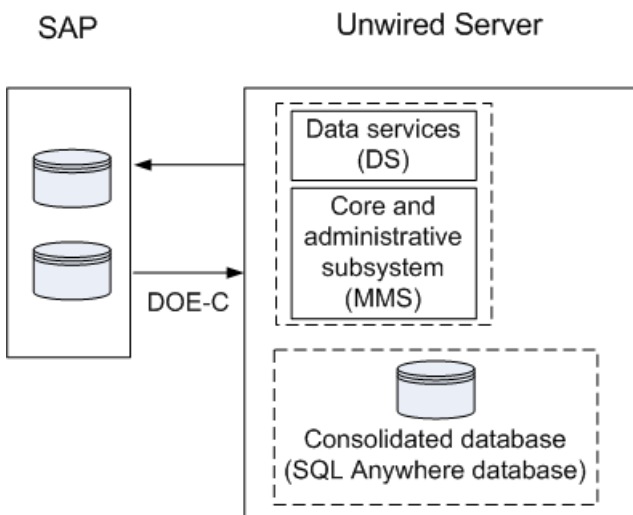
Debugging DOE-C End-to-End Process Flow

Troubleshooting tools for debugging end-to-end processing problems using a DOE connection and SAP data source.

High-level Process Flow

This section describes high-level process flow when DOE-C is configured with Sybase Unwired Platform.

Figure 6: DOE-C Process Flow



Troubleshooting Options

This section describes the options available for diagnosing process flow problems.

SAP-side Debugging

- Verify the SAP data source is running.
- Verify the SAP data source privileges have been granted properly.
- Verify credentials are correct.

- Check for error messages.

DOE Connector (DOE-C) Debugging

- Verify the DOE-C connection is operational.
- Verify credentials are correct.
- Check for error messages.

Sybase Unwired Platform-side Debugging

- Verify Unwired Server is up and running.
- Verify the DOE-C connection is configured properly.

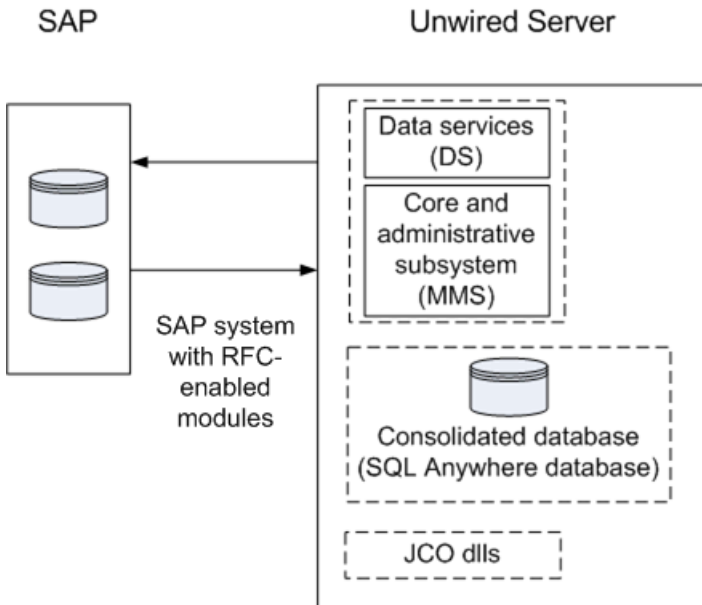
Debugging JCO End-to-End Process Flow

Troubleshooting options for debugging end-to-end processing problems using a JCO connection to an SAP data source and Web service to expose SAP BAPIs.

High-level Process Flow

This section describes high-level process flow when using SAP as a data source.

Figure 7: SAP JCO Connection Process Flow



Troubleshooting Options

This section describes the options available for diagnosing process flow problems.

SAP-side Debugging

- Verify the SAP data source is running.
- Verify the SAP data source privileges have been granted properly.
- Ensure the SAP BAPIs are exposed properly as Web services.
- Verify credentials are correct.
- Check for error messages.

Search *Troubleshooting* for the topic *Verifying SAP Prerequisites*.

SAP JCO Connection Debugging

- Verify the JCO connection is configured properly and is operational.
- Check for error messages.

Sybase Unwired Platform-side Debugging

- Verify Unwired Server is up and running.
- Turn on JCo tracing in the Unwired Server system properties:
 1. In Sybase Control Center for Sybase Unwired Platform, select **Performance Configuration > User Options** for your Unwired Server.
 2. Add the following options to the `-sl java { ... }` section of `mlsrv.ini`:

```
-Djco.trace_level=[0-10]
-Djco.trace_path=[defined_path]
```

The `jco.trace_level` determines the level of detail of tracing you get from the JCO libraries. The `jco.trace_path` determines a directory where trace files get stored. If your SAP-based MBOs fail, turning on tracing can help determine the cause.

- Verify the JCO connection is configured properly. The JCO DLLs and library files must be in the correct location, and they differ for the development environment, and the production environment.

For Unwired Server JCo connections, see the topic *Preparing Unwired Server to Connect to SAP using Java Connectors* in *System Administration*.

For Unwired WorkSpace connections, see the topic *SAP Java Connector (JCo) Prerequisites* in *Sybase Unwired WorkSpace online help*.

Troubleshoot the System

Provides troubleshooting information for the Sybase Unwired Platform system.

Troubleshoot the Installation

Provides troubleshooting information for post-installation configuration problems in Sybase Unwired Platform.

Troubleshooting Host Name Problems

Problem: After installation, you determine that you need to change the host name (machine name); for example, the host name is too long, or includes reserved characters.

Solution: Change the Unwired Server host name using procedures for a single server node, or a node in a cluster. After you change the host name, you must reconfigure Unwired Server to recognize the host name.

After changing the host name, you may encounter temporary connection problems to Unwired Server. This is because the DNS servers may not resolve the IP difference until the connections are refreshed.

Changing Unwired Server Host Name (Single Server)

Change the Unwired Server host name when running as a single server.

1. Uninstall Unwired Server completely as described in the *Installation Guide*.
2. Change the host name using the manufacturer's instructions for the machine.
3. Restart the machine.
4. Install Unwired Server.

Changing Unwired Server Host Name (Node in a Cluster)

Change the Unwired Server host name when running as a node in a cluster.

1. Uninstall the Unwired Server cluster node as described in the *Installation Guide*.
2. Change the host name using the manufacturer's instructions for the machine.
3. Restart the machine.
4. Install the Unwired Server cluster node.

Listeners or Services Could Not be Started Errors

Problem: After installation, Listener could not be started, or Service could not be started messages appear.

Explanation: This may indicate your Sybase Unwired Platform configuration includes duplicate port numbers. This can happen for a variety of reasons; for example, if an application or service is offline during the installation process, the installer does not detect the reserved port number.

Solution: Change the Sybase Unwired Platform component port numbers as needed post-installation, using information in *Port Number Reference*.

Port Number Reference

Change Sybase Unwired Platform component port numbers after installation, if necessary.

Proceed with caution when changing port numbers because the change might impact other configuration files that point to that port. You need to be aware of the default Sybase Control Center port numbers so you do not accidentally use these ports when you change Sybase Unwired Platform ports. You can change some Sybase Control Center default ports, but, in some cases, you should not.

Note: To make Unwired Server port number changes, temporarily stop the other service consuming those ports. Use Sybase Control Center to make the changes, then restart Unwired Server.

Port	Description	Default Port	Instructions for Changing
OpenDS LDAP server	Developer Editions only. Port number on which the OpenDS LDAP server listens for requests. The default administrator password for the LDAP server is secret .	10389 4444 for administration (not used by Sybase Unwired Platform)	Edit the <code>ds-cfg-listen-port : 10389</code> property in <code><Unwired-Platform_InstallDir>\Servers\UnwiredServer\OpenDS\config\config.ldif</code> . Note: If you change the port, remember to change the edit Provider URL property.
Data tier (CDB) server	Port number for the data tier that manages transactions between the enterprise information system and mobile devices.	5200	Do not change the CDB port.

Port	Description	Default Port	Instructions for Changing
Management ports	IIOP port number on which the Unwired Server listens for requests.	2000 2001 for secure management (disabled by default)	<p>Configure in Sybase Control Center by expanding the Servers > <ServerName> folder and selecting Server Configuration. In the General tab, select the Communication Ports subtab and enter a new management port or secure management port, as required.</p> <p><i>See Sybase Control Center online help > Configuring > Configuring Unwired Platform > Unwired Server > Server Properties > Configuring Security Profiles > Management Ports.</i></p>
Data change notification (DCN)	Port number on which Unwired Server listens for DCN events.	8000 for HTTP 8001 for HTTPS	<p>Configure in Sybase Control Center by expanding the Servers > <ServerName> folder and selecting Server Configuration. In the General tab, select the Communication Ports subtab and enter a new DCN port or secure DCN port, as required.</p> <p><i>See Sybase Control Center online help > Configuring > Configuring Unwired Platform > Unwired Server > Server Properties > Security Profiles.</i></p>

Port	Description	Default Port	Instructions for Changing
<p>Replication-based synchronization (RBS) and messaging-based synchronization (MBS)</p>	<p>Port numbers on which Unwired Server synchronizes data between the enterprise information system and mobile devices.</p> <p>MBS port uses a proprietary encryption method, so MBS communication is always encrypted.</p>	<p>2480 for RBS 5001 for MBS</p>	<p>Configure in Sybase Control Center by expanding the Servers > <ServerName> folder and selecting Server Configuration. In the Replication or Messaging tab, select the Synchronization Listener sub-tab and enter a new synchronization port, as required.</p> <hr/> <p>Note: If there is a conflict for port 2480 or 2481, Unwired Server will not start. In that event, you will not be able to use SCC to modify those Unwired Server ports. To correct the problem, you must temporarily stop the service that uses the conflicting port, then start Unwired Server so you can change the port from the SCC console.</p> <hr/> <p>For RBS, see <i>Sybase Control Center > Configure > Configuring Unwired Platform > Unwired Server > Server Properties > Replication > Configuring Replication-Based Synchronization Properties</i>.</p> <p>For MBS, see <i>Sybase Control Center > Configure > Configuring Unwired Platform > Unwired Server > Server Properties > Messaging > Configuring Messaging Synchronization Properties</i>.</p>
<p>Advantage Database Server®</p>	<p>Port number for the messaging database.</p>	<p>6262</p>	<p>Port changes are not recommended.</p>

Port	Description	Default Port	Instructions for Changing
Messaging server administration	Port number for the messaging service for Sybase messaging clients.	5100 for administration services	<p>Cannot be changed in Sybase Control Center.</p> <p>Use the <code><<UnwiredPlatform_InstallDir>>\Servers\Messaging Server\Bin\AdminWebServicesTool.exe</code> command line tool to change the messaging service Web service port. This tool has built in online help describing how to use the tool. From the command prompt run:</p> <pre><UnwiredPlatform_InstallDir>\Servers\Messaging Server\Bin>AdminWebServicesTool.exe set=<port> restart</pre>

Port	Description	Default Port	Instructions for Changing
<p>Sybase Control Center</p>	<p>Additional default port numbers of which to be aware, when modifying port numbers.</p>	<p>9999 for default RMI agent port</p> <p>2100 for default JMS messaging service port</p> <p>3638 for default SCC repository database port</p> <p>8282, 8283 for default Web container ports</p>	<ul style="list-style-type: none"> • 9999 – default RMI agent port. The port is set in: <code><<UnwiredPlatform_InstallDir>>\SCC-XX\services\RMI\service-config.xml</code> • 2100 – default JMS messaging service port. The port is set in: <code><<UnwiredPlatform_InstallDir>>\SCC-XX\services\Messaging\service-config.xml</code> • 3638 – default SCC repository database port. The default port is set in: <code><<UnwiredPlatform_InstallDir>>\SCC-XX\services\SccSAData-server\service-config.xml</code> • 8282, 8283 – default Web container ports. The default ports are set in: <code><<UnwiredPlatform_InstallDir>>\SCC-XX\services\EmbeddedWebContainer\service-config.xml</code> <p>Before you make any changes to these files, stop Sybase Unified Agent service. Start the service after you complete the changes. If any of the subsystems fail to start, check the SCC agent .log for error messages.</p>

Port	Description	Default Port	Instructions for Changing
Relay server	Port numbers on which Relay Server listens for requests.	80 for the HTTP port 443 for the HTTPS port	Change the value for the cluster in Sybase Control Center for Unwired Platform. You can then generate the file and transfer it to the corresponding Unwired Server host. <i>See Sybase Control Center > Configure > Configure Unwired Platform > Clusters > Relay Server.</i>
Unwired Platform reserved	Port numbers reserved for internal use by Unwired Platform components	4343 5500 27000 8002 2638	Do not use these special port numbers for any purpose. These ports differ from Windows reserved ports (1-1023). Note: Even if the installer does not detect a conflict at install time, Windows may later use ports in the 1024-64K range for other purposes. Read Microsoft documentation to determine how to reserve Unwired Platform ports. Otherwise, you may experience intermittent problems when starting up platform services due to Windows using these ports for some other purpose at the same time.

Unwired Platform Windows Services

Unwired Platform Windows services run automatically on your host, with many starting up when the host computer is started or when the installation process finishes. Determine what services exist for each runtime component and what dependencies exist among these services.

Depending on the components installed in the cluster you are administering, some of these services may not appear in your list of Windows services; certain data and server components may be installed on different nodes to facilitate redundancy.

Note: Sybase recommends that you only manually start and stop Sybase Unwired Platform services for debugging and troubleshooting purposes.

If you are routinely starting and stopping Unwired Server, you should use Sybase Control Center for that purpose. Sybase Control Center allows you to manage local and remote servers from a single location, and is more efficient than starting and stopping with services or desktop shortcuts.

Component	Service	Description	Dependencies
Unwired Server	Advantage Database Server	The database server that manages the messaging database.	This service must be started before Sybase Messaging Service can be started.
	OpenDS	The default LDAP server for development authentication and authorization. If you are using a provider other than LDAP, you do not need to use this service. Appears in the Developer Edition installation only.	None.
	Consolidated Database (the service name depends on the data tier installation type): <ul style="list-style-type: none"> If the data tier is installed with Unwired Server on the same host: SybaseUnwiredPlatform<host-name>Database<install_number> If the data tier is installed by itself: SybaseUnwiredPlatformConsolidatedDatabase 	The CDB services.	The CDB service must be started before the Unwired Server service can be started.
	SybaseUnwiredPlatform<cluster_name>Server<install_number>	The Unwired Server service.	Depends on the CDB service startup and relay server service (if used).
	Sybase Messaging Service	The synchronization service that facilitates communication with device client applications.	Depends on Advantage Database Server startup.

Component	Service	Description	Dependencies
Sybase Control Center	Sybase Unified Agent <i>X.X</i>	Provides runtime services to manage, monitor, and control distributed Sybase resources. The agent must be running for Sybase Control Center to run.	None.

Cannot Access Device Tab and Web Service Error

Problem: After installation, you cannot access the **Device Users > Device** tab in Sybase Control Center, and the Unwired Server log shows an ADMIN_WEBSERVICE_INVOCATION_ERROR.

Explanation: This may indicate you have a port number conflict for the messaging administration Web services server listener port.

Solution: Change the port number.

1. Navigate to: `<UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\MessagingServer\Bin`
2. At the command line, enter:

```
AdminWebServicesTool.exe set=<port>
```

Where `<port>` is the new port number. Alternatively, enter this to change the port number and restart the listener using the new number.

```
AdminWebServicesTool.exe set=<port> restart
```

OpenDS Is Not Uninstalled

After uninstalling Sybase Unwired Platform, the OpenDS Windows service is not removed.

Go to **Start > Settings > Control Panel > Administrative Tools > Services** and look for the OpenDS service. If the OpenDS service is there, do one of the following:

- Reinstall Sybase Unwired Platform into the same folder as previously installed. This overwrites the existing OpenDS service instead of installing a second OpenDS service.
- Immediately after uninstalling, reinstall Sybase Unwired Platform to the same folder and install only the server component. Then perform the uninstallation again.

If the OpenDS service is not removed during the uninstallation, and you do not follow one of the above procedures, a second OpenDS service (OpenDS-2) is installed when you reinstall Sybase Unwired Platform. The OpenDS-2 service may fail to start, causing authentication to fail.

Error deleting <filepath>\APIdocs skipping unwired server update

Problem: This error displays while running an installation script for a Sybase Unwired Platform release that does not use an installer, for example: `Error deleting C:`

```
\Sybase\UnwiredPlatform\Servers\UnwiredServer\APIdocs  
skipping unwired server update.
```

Explanation: This message may indicate that the installation script cannot run because Sybase Unwired Platform processes are still running. This may happen when multiple EBFs are applied to update Sybase Unwired Platform.

Solution: All processes should be stopped, using the **Stop Sybase Unwired Services** icon, before running the installation script. If you already did stop the processes through the icon, one of the processes may still be running or in the process of closing down.

Try any of the following:

- If you installed one Sybase Unwired Platform installation script to upgrade, then ran a second installation script to upgrade to a later version, processes may still be running. Try waiting about ten minutes between installing the first and second installation scripts, to give all processes a chance to stop.
- Click the **Stop Sybase Unwired Services** icon again.
- Check Windows Service Manager for any Sybase Unwired Platform-related processes still running, or starting up. If you find any, stop them through Services.
- Use Microsoft Services to check the status of Sybase Unified Agent. If Sybase Unified Agent was not shut down, the upgrade could not complete in a timely manner. You may need to stop Sybase Unified Agent, then retry the upgrade.

Client Application Hangs Waiting for Authentication

Problem: The client application hangs on the login screen, while waiting for authentication from Unwired Server. The application eventually times out and returns an authentication error: Failed to obtain `${licenseNumber}license(s) for SUP_BASECLIENT feature from License Server ${server_id}.`

Explanation 1: This message is from SySam and indicates your site is using more client licences than your purchased SySam license allows.

Solution:

1. Check the Unwired Server log, and look for a checking out licenses error, for example: Failed to obtain 20 license(s) for SUP_BASECLIENT feature from License Server `<server_id>`.
2. Check your license file and find the license type. This file sample shows that only 10 licenses are allowed (CP=10):

```
INCREMENT SUP_BASECLIENT SYBASE 2011.03312 31-mar-2011 uncounted \  
VENDOR_STRING=PE=EE;LT=CP HOSTID=00-1A-A0-36-D4-3F  
ISSUER="CO=Sybase, \  
Inc.;V=1.5;AS=A;MP=365;CP=10;" ISSUED=20-jan-2010 NOTICE="For \  
Sybase Inc. Internal Testing Use Only" SN=12727 TS_OK \  
SIGN2="177C E3EA A4AF 4D7F F470 D299 E7B2 215F 8EBB 76A3 0717 \  
FC2F ED9B A795 BE18 05FB 1871 6352 46D8 1FB7 EAE4 EDB1 CF70 \  
CF6E 970A 6F51 2121 8044 57D9 B87F"
```


This example indicates there is a mismatch between the number of client licenses (20) and the number of Unwired Server licenses allowed (10), so Unwired Server does not authenticate.

3. Reconfigure the license by using `license.bat`, for example: `license.bat EE CP 10`
4. Contact Sybase to correct the license problem.

Explanation 2: This could occur when more device clients are activated and used, than the number of client licenses available.

Solution:

1. Check the Unwired Server log, and look for a checking out licenses error, for example: The number of registered devices has reached the maximum limit for your license..
2. Contact Sybase to purchase more licenses. Once purchased, run the `license.bat` file again.

Explanation 3: This could occur if you have not generated server or client licenses correctly.

Solution:

1. Check your `sup.properties` file to see your license configuration:

```
license.product.edition=EE
license.type=CP
client.licenses=0
```

In this example, 0 indicates there are no client licenses.

2. Check the *Sybase Unwired Platform Installation Guide* for information about generating and downloading license files. See the section starting with: <http://infocenter.sybase.com/help/topic/com.sybase.infocenter.dc00838.0152/doc/html/eka1250809241052.html>.
3. Contact Sybase to correct persistent licensing problems.

Explanation x: This message is from SySam and indicates your site is using more client licences than your purchased SySam license allows.

Solution:

1. Check your `sup.properties` file to see your license configuration:

```
license.product.edition=EE
license.type=CP
client.licenses=-1
```

The `client.licenses` parameter refers to the number of licensed devices. A value of -1 indicates there is no limit to the number of devices.

2. Check the Unwired Server log, and look for a checking out licenses error, for example: Failed to obtain 2147483647 license(s) for SUP_BASECLIENT feature from License Server 27001@mmdfwmap299.mm.local.

This example indicates there is a mismatch between the number of client licenses and the number of Unwired Server licenses, so Unwired Server does not authenticate.

3. Check the license folder and find the license type. For example, you might find the type is ED (and not EE). This means the license configuration must be reset in order for Unwired Server to check out the right client licenses.
4. Contact Sybase to correct the license problem.

Unwired Server or RSOE Startup Problems

Problem: Problems, such as Unwired Server or RSOE failing to start, are encountered without any obvious error messages.

Explanation: This may occur if the Unwired Server host machine is running intrusion detection software, such as the McAfee Host Intrusion Prevention service. This may cause a connection from the CDB to the Unwired Server node to be flagged as an attack, and the CDB host to be added to the list of Blocked Hosts, preventing communication between Unwired Server and the CDB. The Unwired Server log reports connectivity problems with the CDB, and startup failure problems similar to:

```
YYYY-MM-DD HH:MM:SS.SSS ERROR MMS Thread-55
[com.sybase.djc.log.SystemExceptionLog]
com.sybase.djc.DataStoreException was thrown by method
com.sybase.djc.server.PartitionLock.updateLock(boolean)
\ncom.sybase.djc.DataStoreException: Connection Failed:
java.sql.SQLException: JZ006: Caught IOException:
java.net.SocketException: Software caused connection abort:
connect ...
```

Solution 1: You may want to set up some exclusion rules to allow the required communication between the CDB and Unwired Server nodes. Use the following information to formulate the exclusion rules; rules may vary depending on the intrusion detection software used.

- SQL Anywhere® to Unwired Server – SQL Anywhere sends UDP packets on port 2638 in response to discovery probes. The JDBC/ODBC client (on Unwired Server) sends a UDP broadcast on 2638 outbound to all servers on the local network(s) with the name of the database service to which the client is trying to connect. If a SQL Anywhere hosting the given database sees that UDP broadcast request, it responds with a UDP targeted directly back to the host that sent the broadcast.

The installer also issues a 'dblocate' query, which receives UDP responses back from every SQL Anywhere on the local network. Unwired Server uses that to filter out < *cluster-name* > _clusterdb responses to form a list of reachable Sybase Unwired Platform clusters to populate the dropdown list for subsequent installs where the user is adding a new Unwired Server to an existing cluster. If Sybase Unwired Platform is installed in a cluster, the MobiLink™ services also open connections among each other to coordinate activities of the MobiLink farm. Those connections are on the MobiLink listener ports.

- Sybase Unwired Platform to CDB – uses a JDBC connection through jConnect™ for JDBC™ driver. There are two kinds of protocols between Sybase Unwired Platform and CDB:
 - The MobiLink and CDB connection uses Command Sequence.
 - The jConnect for JDBC connection uses TDS.
- RSOE to Relay Server – uses either an HTTP and HTTPS connection.
- Unwired Server to Unwired Server – for administration communication uses:
 - Administration communication (MMS to MMS) uses IIOP protocol.
 - Administration communication (MMS to Mobile Office service) uses IIOP protocol.
- Unwired Server to Advantage Database Server – uses an OLE DB connection through ADO driver.

Solution 2: If your RSOE upgrade failed, you may need to take manual steps to add the RSOE definition and the backend server information to the cluster database. Contact technical support for assistance.

Troubleshoot Deployment

Provides troubleshooting information for deployment problems.

Filter Class Files Not Copied to Deployment Target

Problem: You deploy a mobile business object to the master Unwired Server but the required custom filter class files did not get copied as expected.

Solution: Package your filter class in a deployment unit using Sybase Unwired WorkSpace.

To locate the name of the primary server, look at the server list in Sybase Control Center. The primary server in a cluster is labeled as "primary" in the left navigation pane.

Troubleshoot Synchronization

Provides troubleshooting information for problems that can occur with embedded synchronization components, such as the consolidated database, the remote device database, or the synchronization server.

Synchronization Failed while Downloading Data

Problem: You receive a synchronization failure message while downloading data.

Explanation: This may indicate a deadlock problem with threads in the Unwired Server database used to consolidate enterprise information system (EIS) and device client data. You can confirm this problem by reviewing the exception details and looking for this text in the Unwired Server log file:

[Sybase][ODBC Driver][SQL Anywhere]All threads are blocked

Solution: Set the number of concurrent threads for the consolidated database server to a value that is higher than the number of work threads configured for the synchronization server. To do this, click **Server Configuration** in the left navigation pane of Sybase Control Center for Unwired Server. Change the database server's thread count property as documented in *System Administration*.

Multi-threaded Delete Failures Not Logged

Problem: If you have two threads that try to delete the same record, and the first thread successfully deletes the record, but the second one fails because that record is now removed from the server, no failure message appears in the Unwired Server log.

Solution: None. An attempt was made to delete a record that no longer exists.

Device Synchronization Returns no Data

Problem: Improper syntax in generated code calls causes device synchronization to return no data.

Explanation: This may originate from a misconception about the syntax of generated code calls. In this code example, the deployment_unit file, customer_list_unit.xml, used to autogenerate the class objects, contains this section:

```
<synchronization-parameter name="city" type="string?"/>
```

This indicates that the class object created from the XML requires "city" to be specified prior to syncing with the object.

Solution: Research the mobile business object (MBO) design in Eclipse.

1. Make sure that the synchronization parameter is mapped to an attribute:
 - a) In MBO properties, from the Synchronization tab, select a sync parameter.
 - b) Select an attribute for the "map to" column.
2. After making the change, regenerate the classes, and resynchronize the MBO on the device.

Errors When Reusing Primary Keys

Problem: Reusing an enterprise information system (EIS) primary key generates an error. If you insert a record that has the same primary key that has been deleted, the Unwired Server fails to retrieve the inserted data.

Explanation: Make sure the EIS does not reuse primary keys. For example, if a record with primary key value 1 is deleted, make sure records inserted do not use 1 as the primary key value.

For a database EIS, you can define the primary key column as auto-increment, so when a new record is inserted, the primary key is always assigned a new value.

Troubleshoot the CDB Cache

Provides troubleshooting information for problems that can occur with the consolidated database.

Primary Key Record Missing

Problem: Two records with similar primary key names should exist but only one appears in the cache.

Solution: Check to see whether your datasource is configured as a case insensitive repository. Unwired Platform requires that the datasource be case insensitive. This may require that you change at least one primary key name so they use different characters. Without these changes the first primary key data is overwritten by the second, and this issue will be propagated to client databases when the user synchronizes.

Data Caching Problems

Problem: A data caching problem is detected, or data caching is not operating as expected.

Symptoms may include:

- User 1 downloads the data of User 2 – this might happen if only one version of the data is cached because:
 - Two different CSI users are using different runtime credentials to access data from the backend, and the backend returns data based on the authenticated user.
 - The coherence window is larger than the interval in which the users are syncing.
- All users end up with data on the device that was originally cached by demand of a different device – this might happen if a customized download cursor is used by multiple CSI users using different load parameter values (via different sync parameters).
- If a backend returns data based on an argument that is not filtered, because it is not an attribute of the MBO, all users will download that data by design. If you want to prevent this from happening, propagate the argument into an attribute, so that it can be used as a filter later.

Solution: Implement data partitioning in your MBO design. To enable partitioning of data by user for an MBO located in an cache group with a on-demand refresh policy:

1. Model an additional synchronization parameter and load parameter onto the MBO that requires partitioning, and fill the load parameter with the modeled sync parameter. The sync parameter should be filled with the implicit "username" personalization key.
2. Propagate the artificial load parameter into an artificial attribute of the MBO which must be marked as part of the composite primary key of that MBO.

3. Map the synchronization parameter to the attribute into which the load parameter was propagated.

Troubleshoot Security, Logins, and Authentication

Provides troubleshooting information for security, login, and authentication problems.

Cannot Log in to Unwired Server

Problem: Log-in issues prevent Unwired Server administration.

Solution 1: You may be expecting single sign-on capability, when the same security mechanism has not been configured for both Unwired Server and Sybase Control Center. To ensure the same credentials are used in all components, Sybase strongly encourages you to configure a common security provider for Unwired Server, Sybase Control Center, and Afaia. LDAP and the native operating system (OS) are the only providers that can be used by all Unwired Platform components.

Solution 2: You may have implemented the security provider or server configuration incorrectly, and as a result, cannot log in to Unwired Server. This typically requires that you manually edit a combination of the embedded properties files to correct the problem. Only manually edit the files under the advisement of Sybase technical support.

User Permissions Are Not Correct

Problem: User role permissions are not correct. For example, there seems to be no distinction in permissions between a user role and the Administration role.

Explanation: This typically indicates that Sybase Unwired Platform roles have not been mapped to Sybase Control Center roles, so default Sybase Unwired Platform roles are used.

Solution: Complete role mapping as described in:

- *System Administration > Security Administration > Security Layers > User Security Setup > Security for Administration Users*
- *Sybase Control Center for Sybase Unwired Platform > Configure > Configuring Sybase Control Center > Authorization > Mapping Sybase Control Center Roles to LDAP or OS Groups*

Configuration file locations:

- Sybase Control Center CSI configuration files:
 - `<UnwiredPlatform_InstallDir>\SCC-3_0\conf\csi.properties`
 - `<UnwiredPlatform_InstallDir>\SCC-3_0\conf\roles-map.xml`
- Sybase Unwired Platform CSI configuration file:
 - `<UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\UnwiredServer\Repository\CSI\conf\default.xml`

- `<UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\UnwiredServer\Repository\CSI\conf\role-mapping.xml`

Troubleshooting Single Sign-On

Provides troubleshooting information for problems that can occur when implementing single sign-on (SSO) for SAP enterprise information systems (EIS).

- **When testing, the error `No suitable SAP user found for X.509-client certificate` is logged on the client and Unwired Server when testing SSO with an X.509 certificate on a mobile application client** – during certificate generation, make sure that the user name under which the `dbsvr11.exe` process (Unwired Server) runs, is the same as the user name who generates the credential (`cred_v2`) file. For example, in the Processes tab of Windows Task Manager, the User name under which `dbsrv11.exe` runs is SYSTEM. In this case a SYSTEM user must also generate the certificate.

See this SAP Community Network link <http://forums.sdn.sap.com/thread.jspa?threadID=1756876>.

- **When testing, the error `RFC_ERROR_LOGON_FAILURE: No suitable SAP user found for X.509-client certificate` is logged on the client and Unwired Server when testing SSO with an X.509 certificate on a mobile application client.** –
 1. If using X.509 certificate authentication, remove the JCo properties `jco.client.passwd` (Logon User) and `jco.client.user` (Password) defined for the SAP connection profile in Sybase Control Center.
 2. Regenerate and redeploy the Workflow application and associated MBOs.

Note: You need to remove the SAP connection properties from Unwired Workspace.

- **Application login failures, authentication failures, and operation replay failure error messages** – any of these failures may be caused by the user's password changing in the SAP server. In some cases you may need to create new connection profiles.

For security reasons, the exact nature of the login failure is not returned to the client. Look at the Unwired Server log for details.

- **Using the `createcert` utility to create X.509 certificates to use in SSO testing** – Unwired Platform includes a SQL Anywhere installation that includes the `createcert` utility for generating X.509 certificates.

See *System Administration > System Reference > Command Line Utilities > Certificate and Key Management Utilities > Certificate Creation (createcert) Utility* or the SQL Anywhere documentation for details.

- **Unexpected application behavior when you change credentials in an application** – when you change a password in the client application, it changes only the password which is sent to the SAP server. To change the password that the SAP server accepts, change it at the EIS.

You can import any valid certificate into the device's certificate store, but if the SAP server rejects it, an error is returned to the client.

Troubleshoot EIS Connections

Provides troubleshooting information for enterprise information system (EIS) connection problems.

DOE-C or Messaging Connection Test Errors

Problem: Test of connection properties for messaging or SAP Data Orchestration Engine connector (DOE-C) package fails.

Explanation: The table below lists several common error messages and resolutions.

Error	Package type	Likely cause and resolution
Ping failed	Messaging DOE-C	Server at Service Address is not online. Start up that server.
Unauthorized	Messaging DOE-C	Username or Password are incorrect. Correct the credentials.
Read timeout	DOE-C	Listener URL is incorrect. Remove and redeploy ESDMA package.

Data Source Performance Errors

Problem: Response time for client synchronization degrades, or Unwired Server reports timeout errors while obtaining connections.

Explanation: You may see playback and timeout exceptions reported when some queries are executed:

```
ResourceMonitorTimeoutException: data  
source: <datasource_name> ).
```

You may see errors that indicate client synchronization performance is degraded:

```
Client receives an error code -85  
(SQL_COMMUNICATIONS_ERROR)).
```

These errors indicate that the connection pool size is too small for your user base, and that resources are not managed sufficiently.

Solution: Increase the connection pool size with the `maxPoolSize` property. This value change increases the number of connection pools available to users on your network. You can also increase the value for `maxWaitTime` to increase the timeout value used by the data source.

You set this in the data source connection. In Sybase Control Center, navigate to **Domains** > <Domain> > **Connections**, then select the connection and click **Properties**. The property value can be modified or added as needed.

You can modify connection properties either from Sybase Control Center by clicking **Connections** on the navigation pane, or you can manually modify the ANT file in `<UnwiredPlatform_InstallDir>\Unwired Platform\Servers\UnwiredServer\config\default-data-sources.xml`.

After changing this file, run `configure.bat default-data-sources` to implement changes.

Note: Run `configure.bat default-data-sources` only if you manually modified `default-data-sources.xml`.

SAP Issues

Provides information about issues that may arise when connecting to an SAP® enterprise information system (EIS).

Verifying SAP Prerequisites

You can avoid many SAP issues by meeting the basic prerequisites required to integrate your SAP system with the Sybase Unwired Platform development environment.

Server prerequisites

Check the following for each Unwired Server in the production configuration:

- Verify that the SAP version to which you are connecting is supported. Check the *Sybase Unwired Platform Installation Guide* for supported versions.
- Verify you downloaded the correct versions of `sapjco.jar`, `librfc32.dll`, and `sapjcorfc.dll` from the SAP Web site. Also verify that you installed the files correctly in the installation directory. Search *System Administration* for the topic *Preparing Unwired Server to Connect to SAP using Java Connectors*. These files are critical for establishing connection.
- On Windows 7, Windows Vista, and Windows Server 2008, verify you downloaded the correct versions of `msvcr71.dll` and `msvcpr71.dll` from the SAP Web site. Also verify that you installed the files correctly in the `<WINDOWS_HOME>\System32` directory. See *System Administration* for the topic *Preparing Unwired Server to Connect to SAP using Java Connectors*.
- If you are operating in a clustered production environment, you must also enable SAP mobile business objects to connect to an SAP R/3 system that uses a router. Search *System*

Troubleshoot the System

Administration for the topic *Preparing Unwired Server to Connect to SAP using Java Connectors*.

- If you are using a connection pool, make sure it is configured correctly. Search *System Administration* for the topic *EIS Data Source Connection Properties Reference*.
- If you are using the SAP Data Orchestration Engine Connector (DOE-C), make sure it is configured correctly. Search *System Administration* for the topic *EIS Data Source Connection Properties Reference* and its SAP-related sub-topics.
- If you are using Single Sign-on (SSO), make sure you are configured correctly for the type of credential you are using—X.509 certificate, or SSO2 Token. Search *System Administration* for the topic *Single Sign-on for SAP* and its related topics.

Development prerequisites

Check the following for each Unwired Server in the development environment:

- Verify that the SAP version to which you are connecting is supported. Check the *Sybase Unwired Platform Installation Guide* for supported versions.
- Verify you downloaded the correct versions of `sapjco.jar`, `librfc32.dll`, and `sapjcorfc.dll` from the SAP Web site. Also verify that you installed the files correctly in the installation directory. Search *Sybase Unwired WorkSpace - Mobile Business Object Development* for the topic *SAP Java Connector (JCo) Prerequisites*.

Connecting or Pinging to SAP Generates an Error

Problem: After creating an SAP connection profile, an error is generated when you try to either connect to or ping SAP.

Solution: You may be missing required .DLL files. Windows Vista, Windows 7, and Windows Server 2008 must have the `msvcp71.dll` and `msvcr71.dll` in the Windows system directory for SAP to work correctly with Sybase Unwired Platform. Download these .DLL files from the SAP Web site at <http://service.sap.com/connectors> and place them in your Windows system directory.

For the production environment, see *System Administration > Component Deployment > EIS Connections > Preparing Unwired Server to Connect to SAP using Java Connectors*

For the development environment, see *Sybase Unwired WorkSpace - Mobile Business Object Development > Configure > Configure - Eclipse Development Environment > Creating a Data Source Connection Profile > Creating an SAP Connection Profile > SAP Java Connector (JCo) Prerequisites*.

Correcting Non Fatal Errors During Preview

Problem: Nonfatal "org.eclipse.jface" error message intermittently appears when you preview an SAP MBO.

Solution: Retry the preview operation and ignore the error message.

Troubleshoot Servers

Provides troubleshooting information for servers used in the Unwired Platform configuration.

Unwired Server

Provides troubleshooting information for Unwired Server.

Troubleshooting Unwired Server Performance Issues

Problem: Unwired Server exhibits performance issues.

Explanation: The performance configuration may not be properly tuned to optimize system operation.

Solution: Ensure that all applicable performance-related settings are properly configured.

- Java heap size – when the garbage collection log indicates high activity and the JVM memory allocation pool heap size remains close to the maximum, you may need to reconfigure the maximum and minimum heap sizes. You can enable garbage collection tracing by setting the "JVM options" property. Set appropriate maximum and minimum heap sizes in the **Server Configuration** node (Performance Configuration tab) of Sybase Control Center. The default minimum heap size is 512MB and the default maximum heap size is 2048MB. See *Sybase Control Center online help > Configure > Configuring Unwired Platform > Unwired Server > Server Properties > General Server Ports > Configuring System Performance Properties*.
- Java thread stack size – if you see a StackOverflowError, you may need to increase the default stack size from 400KB. Set an appropriate Java thread stack size in the **Server Configuration** node of Sybase Control Center. Alternatively, you can also decrease the stack size to save more memory. See *Sybase Control Center online help > Configure > Configuring Unwired Platform > Unwired Server > Server Properties > General Server Ports > Configuring System Performance Properties*.
- Messaging-based synchronization (MBS) – if there are large numbers of MBS clients and response times or queue counts are high, consider the inbound and outbound messaging queue counts. These numbers control the number of message queues used for incoming and outgoing messages between the messaging-based synchronization application and the server. If you increase either of these values, ensure you also increase the maxThreads property of in the <hostname>_iiopl.properties file, located in <UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\UnwiredServer\Repository\Instance\com\sybase\djc\server\SocketListener\.

View messaging queue statistics in the **Monitoring** node of Sybase Control Center to calculate the appropriate values and enter these in the **Server Configuration** node of Sybase Control Center. See *Sybase Control Center online help > Configure > Configuring*

Unwired Platform > Unwired Server > Server Properties > General Server Ports > Configuring System Performance Properties.

- Replication-based synchronization (RBS) – if RBS client application performance does not meet the desired level and the CPU or memory settings are not causing this issue, consider increasing the synchronization thread count and cache size for the RBS server. Choose an appropriate synchronization cache size and thread count in the **Server Configuration** node of Sybase Control Center. See *Sybase Control Center online help > Configure > Configuring Unwired Platform > Unwired Server > Server Properties > Replication > Configuring Replication-Based Synchronization Properties.*
- Consolidated database (CDB) – if the CDB server is slow, consider increasing the CDB thread count to allow for more parallel threads to serve incoming requests from the Unwired Server nodes. See *Changing the Consolidated Database Server Thread Count.* Also consider tuning the number of client connections that are saved in the CDB connection pool. Configure this in the 'default' domain by modifying the 'default' data source pool size property in Sybase Control Center. Repeat this change on all nodes in the cluster.
- Data change notification (DCN) listener threads – if there are a large number of concurrent requests and processing times for those requests do not meet requirements, consider increasing the number of DCN listener threads. Set an appropriate number of listener threads in the **Server Configuration** node of Sybase Control Center. See *Sybase Control Center online help > Configure > Configuring Unwired Platform > Unwired Server > Server Properties > General Server Ports > Configuring Communication Port Properties.*
- Server host memory – if you are running Unwired Server on a host with 4GB of memory, by default, 2GB of memory is allocated to applications, such as Unwired Server. If performance issues persist after you have adjusted the aforementioned settings, consider starting Windows with a "/3GB" switch. See [http://technet.microsoft.com/en-us/library/bb124810\(EXCHG.65\).aspx](http://technet.microsoft.com/en-us/library/bb124810(EXCHG.65).aspx).
- Platform connections – for JDBC connections, consider setting the maximum connection pool size to restrict the size of the pool. You can set this value in Sybase Control Center. See *Data Source Performance Errors* or *Sybase Control Center for Sybase Unwired Platform > Configure > Configuring Unwired Platform Connections.*
- Sybase Unwired Platform connection settings to the backend EIS – for JDBC connections to the backend database, you can add options such as `maxPoolSize=<no. of pool size>` to the connection cache property files, located in `<UnwiredPlatform_InstallDir>\Unwired Platform\Servers\UnwiredServer\Repository\Instance\com\sybase\djc\sql\DataSource*.properties`.

For expanded performance tuning details, especially in regards to synchronization performance, see *Synchronization Model Performance Tuning* in the *System Administration* guide.

For more information, see *System Administration guide > Environment Setup > Unwired Server > Server Performance Tuning.*

Troubleshooting Device Tracking in a Cluster

Problem: The server logs for primary and secondary servers list several -10117 stream errors (also known as pull-based notification for Replication-based Sync).

Solution: The Unwired Server consolidated database has issues with tracking devices and light weight polling (LWP) in a clustered environment. You must disable device tracking when LWP is used by reconfiguring all servers in your cluster.

Synchronization Fails With Exception

Problem: Some applications fail to synchronize, and report either an uncaught exception for `waitForKeyGenerator` when calling the **subscribe** method, or an `OutOfMemoryException`.

Explanation: This may happen if Unwired Server does not send the key generator row in response to a login request from the client within 20 seconds of the client's subscribe request. This is typically because the mobile business object is handling a large amount of data. It can also happen if there is a connection problem with the enterprise information system (EIS) that slows response time.

Solution: Try one or more of these:

- Increase the Unwired Server `Max Heap Size`. Search for the *Configuring System Performance Properties* topic found in both *System Administration* and the *Sybase Control Center* online help.
- Redesign the mobile business object (MBO) to handle less data in a transaction.
- Develop custom logging to better trap error messages for more details.

Unwired Server Startup Problems After Changing System Time

Problem: Encountered problems starting up Unwired Server after changing the system time.

Explanation: This can happen if you change the system time to an earlier time setting. If you check the Unwired Server logs you may see messages similar to:

```
YYYY-MM-DD HH:MM:SS.SSS ERROR MMS Thread-8
[com.sybase.sup.server.lm.LicenseManager] msgId: 131160, message:
System clock has been set back.
```

1. Revert the machine to its original time.
2. Uninstall Sybase Unwired Platform.
3. Reset the time if desired.
4. Reinstall Sybase Unwired Platform.

Version Mismatch Detection

The `isCompatible` method prevents version mismatch between incompatible versions of Unwired Workspace and Unwired Server.

Examples include:

Troubleshoot the System

- A later version of Unwired WorkSpace connecting to an earlier version of Unwired Server.
- An earlier version of Unwired WorkSpace connecting to a later version of Unwired Server.

If the two are incompatible, an error message is generated. For this version of Unwired Platform, when Unwired WorkSpace attempts to connect to an older Unwired Server version from a connection profile, the error "UnsupportedOperationException" displays and connection fails.

```
boolean isCompatible(string server_host, int iiop_port, int client_version)
```

Unwired Server responds with true or false based on the the version identifier passed from the client and is checked for support of that version. This method works with any Unwired Server in a cluster, without the need to specify the host and port of the primary.

Relay Server

Provides troubleshooting information for Relay Server or Sybase hosted Relay Server.

Failed to Attach to Relay Server Shared Memory

Problem: Failed to attach to the relay server with the message "<rshost> failed to attach to Relay Server 'ias_relay_server_host' shared memory."

Explanation: This message may appear if you try to update Relay Server configuration using the command `rshost.exe -u -qc -f rs.config`.

Solution: Remove the `-u` option, since the server is not up and running. The `-u` option updates the configuration of a running relay server.

HTTP_BAD_STATUS_CODE Error

Problem: The HTTP_BAD_STATUS_CODE error appears on the console or in the log files.

Solution 1 (iOS):

1. For iOS devices, navigate to the Settings application on the device.
2. Click the corresponding application tab.
3. Confirm that the connection information is correct

Solution 2 (Windows Mobile):

1. For Windows Mobile devices, check the connection parameters in the `ApplicationInit.cs` code to make sure the relay server connection configuration is correct.
2. Confirm that the `rshost.exe` is running under the system user ID.

SOCKET_HOST_NAME_NOT_FOUND Error

Problem: The SOCKET_HOST_NAME_NOT_FOUND 10050 error appears on the console or in the log files.

Solution 1 (iOS):

1. Navigate to the Settings application on the device.
2. Click on the corresponding application tab.
3. Confirm that the connection information is correct.

Solution 2 (Windows Mobile):

1. For Windows Mobile, if you are using an emulator for testing, confirm that you have synchronized the emulator first.
2. Confirm that `dblgenn11.dll` is in your `C:\Inetpub\wwwroot\ias_relay_server\server\` folder.
3. Confirm that the RSOE version is less than or equal to the relay server version.

Connection Dropped Due to Lack of Network Activity

Problem: The error message Connection was dropped due to lack of network activity appears on the console or in the log files.

Explanation: This message indicates that the relay server outbound enabler (RSOE) sent a PING to Unwired Server synchronization service to determine whether it is active.

Solution: This message is informational and can be ignored. If you want to suppress the message, you can remove the `-v+` switch from the Unwired Server.

Sybase Hosted Relay Server

Provides troubleshooting information for Sybase hosted Relay Server.

Sybase Hosted Relay Server Fails to Connect

Problem: Sybase Hosted Relay Server fails to connect in a development or test environment.

Solution: Check the log files for information. There are several possibilities.

- The Farm ID and Server IDs may differ between the Relay Server outbound enabler (RSOE) and the Sybase hosted Relay Server. Ensure that these IDs match exactly by checking `<UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\UnwiredServer\config\relayserver.properties`. If they do not, delete the values of `relayserver.farm_name` so that it matches the one configured in the Sybase Hosted Relay Server.
- The messaging service may not be configured properly, or may not be configured at all. Check the values in `<UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\UnwiredServer\config\relayserver.properties` to make

sure `relayserver.imo.farm_name` and `relayserver.imo.token` are configured properly. Add them or correct them as needed.

- The RSOE and configuration changes may not have been registered, meaning that additional files have not been updated with the changes (such as `rs.config`, `startrsoe.bat`, and `stoprsoe.bat`). Register the RSOE and configuration changes by running `<UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\UnwiredServer\bin\regRelayServer.bat`.
- The Relay Server outbound enabler may need to be restarted. To do so, restart Unwired Server. Unwired Server and messaging service should connect to the Web hosted relay server.

Messaging Server

Provides troubleshooting information for the embedded Unwired Server messaging service.

Messaging Service Startup Failure

Problem: The installer creates a service that is defined by the user performing the installation. When Windows forces a domain password change, the Unwired Server messaging service fails to start, because the former domain password is still used.

Solution: Change the password for the service by:

1. In the Windows Control Panel, open the Services pane.
2. Right-click Sybase Messaging Service, and choose **Properties**.
3. In the **Log On** tab, configure properties for the host computers user account and ensure the newly changed domain password is used.
4. Click **Ok**.
5. Right-click Sybase Messaging Service, and choose **Restart**.

Unable to Register the MobiLink Server

Problem: Starting Unwired Server causes registration problems for Messaging Server.

The Unwired Server error log displays an entry similar to:

```
2010-07-27 09:37:31.205 ERROR Mobilink Thread-146
  [com.sybase.ml.sup.Logger] [-10002] Consolidated database server
or ODBC
  error: ODBC: [Sybase][ODBC Driver][SQL Anywhere]Index 'ml_server
UNIQUE
  (name)' for table 'ml_server' would not be unique (ODBC State =
23000,
  Native error code = -196)
2010-07-27 09:37:31.205 ERROR Mobilink Thread-146
  [com.sybase.ml.sup.Logger] [-10294] Unable to register the
MobiLink
  server with name '<serverName>'
```

Explanation: When Unwired Server runs in a shared server farm, every member in the farm must have a unique registration through the consolidated database table `ml_server`. If

Unwired Server is not stopped gracefully, the row for the server is not removed from the **ml_server** table. This could happen if Unwired Server is shut down abruptly while a mobile client is synchronizing. When Unwired Server is restarted, the Messaging Server registration is not unique, and cannot be completed.

Solution:

1. Connect to the consolidated database using **dbisql**.
2. Manually remove the row corresponding to your server. An example row:

```
server_id,name,version,connection_info,instance_key,start_time,li
veness 9,'ML Server
name','100.0','http(host=<supHostName>;port=2480)',
0xad53f35e04086778623aa56f32728a313f1972b68b8363c86b501144af7b320
6,'2010-07-27 14:54:10.298','2010-07-27 18:29:13.866'
```

3. If the problem persists, try restarting Unwired Server.

Troubleshoot Clusters

Provides troubleshooting information for cluster problems related to Sybase Unwired Platform.

Advantage Database Server Cannot Authenticate

Problem: After installing Unwired Platform, with a high-availability data tier cluster (using Microsoft Cluster Service or Failover Clustering), this error message displays in the event viewer: 7078: The Advantage Database Server cannot authenticate the user. Make sure the name and password are correct..
axServerConnect AdsConnect.

Explanation: This typically indicates that the Unwired Server messaging service cannot locate the shared cluster storage resource that houses database files for the data tier.

Solution:

1. On a data tier host, check the following registry key value to find the location of the `adsserver.ini` file: `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\Advantage\Configuration\TPS_LOGS`. By default it is set to: `C:\`
2. If the `adsserver.ini` file does not exist, create it in the location cited, on each data tier node in the cluster.
3. Add the following lines to the `adsserver.ini` file, on each data tier host, to identify the physical path to the shared cluster storage resource:

```
[ServerAliases]
<network_share_name>=<local_path>
```

In this example, `Data` is the network share name, and `F:\` is the local path of the share folder on each node.

```
[ServerAliases]  
Data=F:\
```

4. Restart the Advantage Database Server service.

Troubleshoot Workflow DCN Failures

Problem: An error code and message may be returned to the calling party when a workflow data change notification (DCN) fails.

Solution: Resolve the problem and resend the workflow DCN. Unwired Server failures and errors include:

- **Unwired Server down** – HTTP 400 is returned to the calling party
- **Missing DCN parameter, such as cmd=wf** – HTTP 500 is returned to the calling party
- **The workflow DCN message is sent to a non-existent user** – this error message is returned to the calling party:

```
xmlResponse: {"id":"000000392902","success":false,  
"statusMessage":"no workflow message is sent because no  
device is found to send the mail","result":[]}
```

Troubleshoot Sybase Control Center for Sybase Unwired Platform

Provides troubleshooting information for the Sybase Control Center (SCC) used to manage and monitor Unwired Platform.

Using Sybase Control Center to Troubleshoot Unwired Platform

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

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

1. Review the server log – view server errors, warnings, and general information to identify problems. Access the Server node in the left navigation tree of Sybase Control Center to view server log data.
2. Review domain logs – if domain logging is enabled, view domain logs in each Domains > <DomainName>> Log node of Sybase Control Center. Aggregated log data in the console makes domain information readily accessible and actionable.
3. Review monitoring data – access the Monitoring node in the left navigation tree of Sybase Control Center to view monitoring data on the following components of Unwired Platform: replication-based synchronization, messaging-based synchronization, messaging queue, data change notifications, device notifications, packages, users, and cache. Search *System Administration* for *System Diagnostics*.
4. Review device statuses – access the Device Users node in the left navigation tree of Sybase Control Center to view replication- and messaging-based synchronization device and user information.
5. Review package client logs – access the Client Log tab of the Packages > <PackageName> node in Sybase Control Center to view data about client application operations for all devices subscribed to a package. This information allows you to track errors and identify performance issues.
6. Review MBO and operation history – access the History tab for both the MBO and operation nodes of a package in Sybase Control Center to review error history during synchronizations and operation replays.

Collecting Administration Performance Data for Troubleshooting

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

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

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

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

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

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

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

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

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

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

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

Sybase Control Center Management Tier Issues

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

Launching SCC Results in Rounded Rectangle Box or Empty Console Screen

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

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

Solution: Upgrade your Flash Player version to at least 9.0.124 or higher. However, Sybase recommends that you use the most recent available version of Flash Player 10. For more information on software prerequisites, see the *Sybase Unwired Platform Installation Guide*.

Sybase Unified Agent Windows Service Fails to Start

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

The <UnwiredPlatform_InstallDir>\SCC_3-0\log\agent.log shows the following message:

```
2009-11-10 13:35:50,752 [INFO ] [main]
com.sybase.ua.util.process.ProcessHandler.startProcess(395) -
Starting process SQLAnywhere-scc_repository: C:\Sybase
\SCC-3_0\services\ScsSADataserver\sa\bin_windows32\dbsrv11.exe
-n scc_repository -o C:\Sybase\SCC-3_0\services\Repository
\scc_repository.slg -m -qi -qw -sb 0 -gn 100 -gm 500 -zl
-zp -x TCPIP{port=3638} C:\Sybase\SCC-3_0\services\Repository
\scc_repository.db
2009-11-10 13:36:57,037 [ERROR] [main]
com.sybase.ua.services.asa.ASAEngine.startEngine(135) - Failed to
connect to
dataserver engine. ASAEngine: Unable to open a new connection to the
engine.
```

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

Solution 1: Review <UnwiredPlatform_InstallDir>\SCC-3_0\services\Repository\scc_repository.log log for any issues with the database transaction log file during startup. If the transaction log could not be processed, the database cannot start, and consequently nor can the Unified Agent service. Resolve this error by:

1. Creating a backup of <UnwiredPlatform_InstallDir>\SCC-3_0\services\Repository\scc_repository.log.
2. Deleting the <UnwiredPlatform_InstallDir>\SCC-3_0\services\Repository\scc_repository.log file and restarting the Unified Agent service.

Solution 2: Review <UnwiredPlatform_InstallDir>\SCC-3_0\services\Repository\scc_repository.log log for any failures in database recovery. Resolve this error by temporarily configuring the consolidated database to start without a transaction log:

1. Log out of Sybase Control Center.
2. Open <UnwiredPlatform_InstallDir>\SCC-3_0\services\ScsSADataserver\service-config.xml.

3. Locate this line: `com.sybase.asa.server.options`.
4. Append "-f" to the value of this property.
5. Save the changes.
6. Restart the Unified Agent service.
7. Once the database server has processed the incomplete transaction, it will shutdown the database.
8. Confirm the successful transaction in the log file.
9. Revert the configuration of `<UnwiredPlatform_InstallDir>\SCC-3_0\services\ScsSADatasever\service-config.xml`.
10. Restart the Unified Agent service.

Sybase Unified Agent Windows Service Deleted

Problem: the Sybase Unified Agent Windows service was inadvertently deleted, so Sybase Control Center is unavailable.

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

```
UnwiredPlatform_InstallDir\SCC-3_0\utility\ntautostart\release
\uaservice.exe -install
```

Sybase Control Center Fails to Start

Problem: The Sybase Control Center server does not start.

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

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

1. From the command line, verify the host name by running `nslookup<hostname >`.
2. If the DNS server cannot resolve the host name, edit the colocated `<UnwiredPlatform_InstallDir>\SCC-XX\services\RMI\service-config.xml` file:
 - a. Log out of Sybase Control Center.
 - b. Stop the Sybase Unified Agent service.
 - c. Open `<UnwiredPlatform_InstallDir>\SCC-XX\services\RMI\service-config.xml`.
 - d. Locate this line: `<set-property property="address" value="<hostname>" />`.
If the line does not exist, add it under the `<properties></properties>` element in the file.

- e. Change the value from the host name to the IP address of the host computer. If the IP address is already used, ensure it is valid (especially if the IP address has recently been changed).
- f. Restart the Sybase Unified Agent service.
- g. Log in to Sybase Control Center and proceed with your administrative tasks.

Second Sybase Control Center Fails to Start

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

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

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

Login Invalid in Sybase Control Center

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

Solution:

- Verify Sybase Control Center session validity – ensure that the current Sybase Control Center session is active. If the session is frozen or expired, refresh the page or close the browser and try again.
- Verify authentication configuration – ensure that the Sybase Control Center authentication provider configuration is correct, and points to the correct server. See *Security for Administration Users* in the *System Administration* guide for more information.
- Verify LDAP listener port consistency – if you are using LDAP security, ensure that the LDAP server port number is the same in all configuration files. The configuration files you may need to edit vary, depending on license type, or whether Unwired Platform shares the same repository as Sybase Control Center.
 - For development licenses: `<UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\UnwiredServer\OpenDS\config\config.ldif` and check the **ds-cfg-listen-port** property. This property value must match that specified in the Sybase Control Center authentication provider configuration file in addition to Unwired Platform if they share the same repository.
 - For Sybase Control Center authentication: `<UnwiredPlatform_InstallDir>\SCC-XX\conf\csi.properties`, and configuration should point to the correct LDAP host and port with appropriate properties.
- Ensure that Unwired Platform administration roles are correctly mapped to Sybase Control Center roles. Otherwise, you must first log in to Sybase Control Center as the `sccAdmin` (no password) and then, when you authenticate with Unwired Server, log in

again with your Unwired Platform administrator role. For information about logging into Sybase Control Center and authenticating with Unwired Server, search for the Sybase Control Center online help for *Getting Started with Unwired Platform Administration*. For information about mapping roles, see *Security for Administration Users* in the *System Administration* guide.

- Check `<UnwiredPlatform_InstallDir>\SCC_XX\log\agent.log` for any issues with starting various services.

Cannot Access Unwired Platform Administration Nodes

Problem: Unwired Platform navigation nodes are documented but are either not visible or not functional in Sybase Control Center.

Solution: Ensure that you have the correct administration roles assigned, and that your product and version is Sybase Unwired Platform 2.0. Other products like Sybase Mobile Sales or Sybase Mobile Workflow may have visible Unwired Platform features, but may not have all of the same functionality as Sybase Unwired Platform 2.0.

Previous Administrator Credentials Used

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

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

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

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

Browser Refresh (F5) Causes Logout

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

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

Security Error Triggered When Connecting to SCC from Remote Browser

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

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

Administrator Login Passes When Provider Is Not Available

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

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

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

Host Name of Registered Resource Changed But Is Not Updated

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

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

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

Management Issues with Clustered Data Tiers

Problem: if you install Unwired Platform and the consolidated database on MS Cluster, you will receive errors when trying to manage the cluster in Sybase Control Center. This is because MS Cluster uses node switches.

Solution: Replace the current entry for the cluster with a new entry that uses the computer node's hostname or IP address, rather than Unwired Platform cluster's hostname (the default).

Platform Component Monitoring Issues

Review this list of documented issues for platform components monitored by Sybase Control Center.

Monitoring Data Does Not Appear in History Tab

Problem: Monitoring data does not appear immediately in the History tab.

Explanation: The monitoring data is stored in memory to optimize database access, and periodically flushed to the monitoring database.

Solution: Try either of these options:

- Wait for the data to be flushed. The default time period is five minutes.
- Change the flush interval to a smaller value in Sybase Control Center:
 1. In the left navigation pane, select **Monitoring**.
 2. In the right administration pane, select the **General** tab.
 3. Click **Configuration**.
 4. In the flush threshold section, ensure that **Enable flush threshold configuration** is selected.
 5. Select one of:
 - **Number of rows** – monitoring data that surpasses the specified number of rows is flushed from the console display. Enter the desired number of rows adjacent to **Rows**. The default is 100.
 - **Time interval** – monitoring data older than the specified time interval is flushed from the console display. Enter the desired duration adjacent to **Minutes**. The default is 5.
 - **Either rows or time interval** – monitoring data is flushed from the console display according to whichever value is reached first: either the specified number of rows or the specified time interval. Enter the desired rows and duration adjacent to **Rows** and **Minutes**, respectively.
 6. Retrieve the results list using the Sybase Control Center monitoring node.

Previously Existing Monitoring Data No Longer Appears

Problem: Monitoring data that displayed previously no longer appears.

Explanation: By default, monitoring data is preserved in the database for seven days. After that period, the data is removed.

Solution: Change the auto purge setting value in Sybase Control Center. Auto purge clears obsolete data from the monitoring database once it reaches the specified threshold.

1. In the left navigation pane, select **Monitoring**.
2. In the right administration pane, select the **General** tab.
3. Click **Configuration**.
4. In the auto purge section, ensure that **Enable auto purge configuration** is selected.
5. Enter the length of time (in days) to retain monitoring data before it is purged.
6. Restart the server.

7. Retrieve the results list using the Sybase Control Center monitoring node.

Server Tier Administration Issues

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

Server List Not Retrieved

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

Scenario 1: No other error message appears.

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

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

Solution: Manually register the remote server. By default, only Unwired Servers installed to the same host computer are automatically registered with Sybase Control Center. See *Getting Started with Unwired Server Administration* in the Sybase Control Center online help. If you have recently made changes to the environment, for example, by modifying server resource properties (login, password, host name, IP address, or port number), ensure that you reauthenticate after making the changes.

- Jetty caching in Sybase Control Center prevents the console from displaying the server tree. This is indicated by 404 errors in both the console URL and `<UnwiredPlatform_InstallDir>\SCC-XX\services\EmbeddedWebContainer\log\http-service.log` (the HTTP access log).

Solution:

1. Close Sybase Control Center.
2. Stop Sybase Unified Agent Service.
3. Delete the contents of: `<UnwiredPlatform_InstallDir>\SCC-XX\services\EmbeddedWebContainer\container\Jetty-6.1.22\work`.
4. Restart Sybase Unified Agent Service.

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

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

- The security repository (LDAP server) is down or not reachable. In the default configuration (excluding the Deployment Edition), the security repository is the OpenDS LDAP server.

Solution: Check the status of the security repository to ensure that the server is reachable. For the OpenDS LDAP server, check the Services in the Windows Control Panel to ensure

that the service is running and that the server is reachable from Sybase Control Center and the Unwired Server host.

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

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

- You do not have the required administration privileges.

Solution: Ensure that:

1. The administrator login is assigned one of the physical roles attached to the "SUP Administrator" or "SUP Domain Administrator" logical roles in the "admin" security configuration. To change the "admin" role mapping outside of Sybase Control Center, manually edit `<UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\UnwiredServer\Repository\CSI\conf\role-mapping.xml` on all nodes in the cluster.
2. The `<UnwiredPlatform_InstallDir>\SCC-XX\conf\roles-map.xml` configuration for the "SUP LDAP Login Module" and your security repository are being used to authenticate Sybase Control Center users. Ensure that the administration roles configured for the "SUP LDAP Login Module" exist in the security repository.

- The server IP may have changed.

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

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

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

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

Solution: Check `<UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\UnwiredServer\logs\<hostname>-server.log` for details.

- The Sybase Control Center security provider is down or a system condition prevents Sybase Control Center from authenticating the user for administration access.

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

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

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

Unwired Server Fails to Start

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

Solution:

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

Error in Listing Devices and ADMIN_WEBSERVICE_INVOCATION_ERROR in Agent.log

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

If users report problems synchronizing from the client, check for this error message in the Sybase Control Center agent.log file:

```
2009-11-10 17:19:51,718 [ERROR] [btpool1-4]
com.sybase.scc.jmx.ManagedObjectGateway.invoke(?) - Failed to invoke
managed object '82061EE7A2D141E391B46D245EAF0B1E' mbean operation
searchDeviceList2([java.lang.String]):
com.sybase.uep.sysadmin.management.mo.exception.SUPartifactMOExcept
ion: com.sybase.uep.sysadmin.management.mbean.UEPAdminException:
com.sybase.uep.admin.client.AdminException:
ADMIN_WEBSERVICE_INVOCATION_ERROR: java.security.PrivilegedActionExc
eption: com.sun.xml.internal.messaging.saaj.SOAPExceptionImpl:
Message send failed
javax.management.MBeanException:
```

Explanation: There are two possible reasons – the service could not be started due to a service account credential issue, or there is a conflict on the currently configured port for the administration web service.

If you install Sybase Unwired Platform with a domain account with administrator rights, and subsequently change the password, or use a domain account that does not have administrator rights, the Sybase Messaging Service will not start. One way to verify this is by accessing the following URL from the host where Sybase Unwired Platform is installed: `http://localhost:5100/MobileOffice/Admin.asmx`. The default Messaging Service port is 5100, but this may vary depending on your configuration.

Solution 1: Update or provide the username and password:

1. Navigate to **Control Panel > Administrative Tools > Services**.
2. Right-click Sybase Messaging Service, and select Properties.
3. Select the Log On tab, then update or provide the username (with administrative rights), and correct the password.

Solution 2: Make sure the administration Web service is up and running, and correctly configured. Review *Troubleshooting > Troubleshoot the System > Troubleshoot the Installation > Cannot Access Devices Tab and Web Service Error* topic on how to configure the port in case of conflict on existing port.

Starting or Restarting a Remote Server from Sybase Control Center Fails

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

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

Before attempting the following solutions, verify that:

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

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

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

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

- If you cannot resolve the local host name, modify the file on the local instance of Sybase Control Center.
 - If you cannot resolve the remote host name, modify the file on the remote instance of Sybase Control Center.
 - If you cannot resolve both the remote and local host names, modify both files.
1. From the command line, verify the host name by running `nslookup<hostname >`.
 2. If the DNS server cannot resolve the host name, edit the colocated `<UnwiredPlatform_InstallDir>\SCC-XX\services\RMI\service-config.xml` file:
 - a. Log out of Sybase Control Center.

- b. Stop the Sybase Unified Agent service.
- c. Open `<UnwiredPlatform_InstallDir>\SCC-XX\services\RMI\service-config.xml`.
- d. Locate this line: `<set-property property="address" value="<hostname>" />`.
If the line does not exist, add it under the `<properties></properties>` element in the file.
- e. Change the value from the host name to the IP address of the host computer. If the IP address is already used, ensure it is valid (especially if the IP address has recently been changed).
- f. Restart the Sybase Unified Agent service.
- g. Log in to Sybase Control Center and proceed with your administrative tasks.

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

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

Port Conflict Issues

Problem: You have identified a Sybase Unified Agent port conflict.

Solution:

1. Identify the service with the port conflict in `<UnwiredPlatform_InstallDir>\SCC_3-0\log\agent.log`.
2. Use a text editor to open `<UnwiredPlatform_InstallDir>\SCC_3-0\Services\<Servicename>\service-config.xml`.
3. Change the port to an available port number.
4. Save and close the file.

See *Sybase Unwired Platform System Administration Guide > System Reference > Port Number Reference* for more information.

Unexpected Listener Startup or Connection Errors

Problem: You encounter unexpected listener startup or connection errors for Unwired Platform components.

Solution:

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

To do so on Windows XP, navigate to: **Control Panel > Network Connections > Local Area Connection 1 > Properties > General tab > Internet Protocol (TCP/IP) > Properties > General tab > Advanced > Options tab > TCP/IP filtering > Properties**

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

Refreshing Server Configuration Displays Only Partial Updates

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

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

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

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

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

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

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

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

1. Make the first change.
2. Save the configuration.
3. Refresh the configuration.
4. Make the second change.

5. Save the configuration.
6. Restart the server.
7. Refresh the configuration.

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

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

Users Connect with Old Credentials

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

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

Solution: To prevent caching of login credentials or reduce the cache period, decrease the default login cache timeout by changing the `loginCacheTimeout=60` property in `<SUP_HOMW>/Servers/UnwiredServer/Repository/Instance/com/sybase/djc/security/SecurityDomain/default.properties`. Repeat this change on all server nodes in the cluster.

AuthorizationException Displays Instead of Status

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

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

Solution:

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

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

Increasing Messaging Queue Counts Degrades Performance

Problem: Both inbound and outbound messaging queue counts were increased, however, performance degraded as a result.

Description: After increasing inbound and outbound message queue count, the default maxThreads of IIOP socket listener is insufficient.

Solution: Increase the maxThreads of IIOP socket listener by editing the <hostname>_iiopl.properties file (located in <UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\UnwiredServer\Repository\Instance\com\sybase\djc\server\SocketListener\), and restart Unwired Server. The maxThread of IIOP socket listener must be larger than the sum of all nodes needed IIOP thread counts.

Saving Server Configuration Fails Due to Certificate Validation Error

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

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

Package Deployment and Management Issues

Review this list of documented issues for packages deployed or managed from Sybase Control Center.

Exporting or Deploying Large Packages Fails

Problem: You used Sybase Control Center to export or deploy a large package, and it fails.

You can troubleshoot this error by opening the Sybase Control Center <UnwiredPlatform_InstallDir>\SCCXX\log\agent.log file and checking for a message that is similar to this one:

```
exception: java.lang.IllegalStateException: Form too large
```

Explanation: This message means that the package, and not the form, is too large. The Web server that hosts Sybase Control Center cannot manage the data. A number like 273310 indicates the size of the package in kilobytes (that is, 273,310).

Solution 1: Use this solution if you run the Sybase Unified Agent as a service (default).

1. Sybase recommends you create a backup copy of the Windows registry before proceeding.

2. Open the registry editor by running **regedit** at the command line.
3. Locate the Sybase Unified Agent 3.0 registry key: HKEY_LOCAL_MACHINE \SOFTWARE\SYBASE\Unified Agent 3.0.
4. Set the `jvmopt1` property to a value larger than the default, and save the file. The default is 2000000. For example:

```
-Dorg.mortbay.jetty.Request.maxFormContentSize=2048000
```

Solution 2: Use this solution if you do not run the Sybase Unified Agent as a service.

1. Close Sybase Control Center, and stop Sybase Unified Agent using the Windows Services dialog.
2. Open the `uafstartup.bat` file, located at `<UnwiredPlatform_InstallDir>\SCC_3-0\bin\uafstartup.bat`, in a text editor.
3. Set the **maxFormContentSize** property to a value larger than the default, and save the file. The default is 2000000. For example:

```
-Dorg.mortbay.jetty.Request.maxFormContentSize=2048000
```
4. Restart the Sybase Unified Agent (using the updated `uafstartup.bat` file), and reopen Sybase Control Center.

Invalid DOE-C User Error for an SAP Server Connection

Problem: The General tab of a DOE-C package displays an invalid user account error for the Error State property.

Explanation: SAP servers could not authenticate this user with the Username and Password configured for this package.

User names and passwords configured for the connection pool cannot be tested before they are used. Errors are only reported after the connection fails. Errors typically occur during an administrative operation (such as unsubscribing a subscription), or in response to an asynchronous message for a subscription from DOE. On a system with existing DOE-C subscriptions, the initial resynchronization at startup would implicitly test the technical user.

Solution: Check the username and password configured for this user in the Connection Pool configured for the package. If it is incorrect, edit the properties used.

Note: If you change the username or password property of a DOE-C connection, you must reopen the same dialog and click **Test Connection** after saving. Otherwise the error state of this DOE-C package cannot be cleaned up. If you do not click **Test Connection**, the username or password is correct, but the error state of the DOE-C package cannot be cleaned up.

Device and Device User Management Issues

Review this list of documented issues for devices or device users managed by Sybase Control Center.

Wrong Device for Code Error

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

This error occurs when you:

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

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

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

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

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

Solution: Reconnect the emulator by either:

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

User Name of Registered Device Not Displayed

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

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

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

1. In the left navigation pane, click the **Device Users** node.
2. In the right administration pane, click the **Devices** tab.
3. In the table of registered devices, select MBS to display the **Activation User Name** column and look for the device ID.

Internal Server Error When Clicking Device User

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

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

Solution:

1. Validate the error:
 - a. Open `<UnwiredPlatform_InstallDir>\SCC-3_0\log\agent.log`.
 - b. Look for this error: `Caused by: com.sybase.uep.sysadmin.management.exception.ImoWsException: An error occurred loading a configuration file: Attempted to read or write protected memory. This is often an indication that other memory is corrupt.`
2. Validate that the Messaging Server service (obmo.exe) is running.
3. Validate that the Messaging Server Administration Web Service is running:
 - a. Open a Web browser.
 - b. Open `http://localhost:5100/MobileOffice/admin.asmx`.
 - c. Select the **GetDeviceList2** method, then click **Invoke**.
 - d. Check whether a valid XML response returns.
4. If anything in steps 1-3 is unexpected, you may have an installation or configuration issue. Confirm this by:
 - a. Restarting the Messaging Server service.
 - b. Once available, repeat steps 2-3.
 - If you still do not yield an expected result, then reinstall the Unwired Server.

- Otherwise, open Sybase Control Center, and click Device User to try registering a user again.
5. If you still get the same error and same behavior, contact Sybase Support. Ensure you:
- Have a copy of the Messaging Server deviceManagement trace file.
 - Including the information on which specific Admin WS method is called that causes the error, and what parameter value is used.

Troubleshoot Mobile Business Object Development

Provides troubleshooting topics for problems you may encounter while using Sybase Unwired Platform to develop, test, and deploy mobile business objects (MBOs).

See the Validation Rules section for information about validation rules and error messages.

Cannot Ping Unwired Server

Problem: Cannot ping the Unwired Server from the Sybase Unwired WorkSpace development environment.

Explanation: You can use OpenDS only in a development environment or for trial purposes; it is not intended for use with other LDAP providers in a production environment. The Sybase Unwired Platform OpenDS service uses port 10389 for LDAP; if another LDAP provider, such as IBM Lotus Domino *x.x* server, starts on the same machine and uses port 10389, Sybase Unwired WorkSpace cannot successfully ping Unwired Server.

Solution: Use one LDAP service at a time, or change the Sybase Unwired Platform OpenDS port. See the *Port Number Reference* topic for information about ports.

Explanation 2: Verify that Unwired Server is up and Sybase Unwired Platform services are running.

Solution: If Sybase Unwired Platform services are not running, restart the services in the correct order. See the *Starting and Stopping Unwired Platform Server Services* topic in the *Release Bulletin*.

Cannot Access sampledb

Problem: After installation, you cannot access the sample database (`sampledb`).

Explanation 1: If you installed the Deployment Edition, this is expected behavior. You must start `sampledb` manually for Deployment Edition.

Solution: Use the Start and Stop `sampledb` command line utility (**`samledb.bat`**) to start the sample database. See *System Administration for Sybase Unwired Platform > System Reference > Command Line Utilities > Unwired Server Runtime Utilities > Start and Stop `sampledb` Server (`sampledb`) Utility*.

Explanation 2: If you installed a Development Edition, make sure the `sampledb` windows services is installed properly and started. See *System Administration for Sybase Unwired*

*Platform > System Reference > Command Line Utilities > Unwired Server Runtime Utilities
> Start and Stop sampledb Server (sampledb) Utility.*

Data Source Connection Fails After Installing to a New Directory

Problem: After uninstalling Sybase Unwired Platform, then installing to a new directory, a data source connection fails in Unwired WorkSpace, with the message:

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

Explanation: This may happen if you use an old WorkSpace, and a JDBC connection profile. The data source connection may be using the original directory.

Solution:

1. Once Sybase Unwired WorkSpace is installed, switch to a new WorkSpace directory.
2. Or, manually change the driver path by modifying the properties of the connection profile for a New Driver Definition.

MS SQL Server Connection Errors

Problem: When connecting to MS SQL Server, the connection succeeds but the error log receives continuous errors.

Solution: This can happen if the connection includes a nonexistent database name, such as the default name, pubs. When you create the connection, remove the incorrect database name from the connection URL, `jdbc:sqlserver://localhost:1433;databaseName=pubs`, and enter the correct database name.

ODBC Driver Errors

Problem: ODBC driver errors appear when you try to connect to a database from Sybase Unwired Platform installed on a 64-bit operating system.

This can happen if the registry has been changed, so the ODBC driver cannot be found. For example, installing a standalone version of SQL Anywhere may cause registry changes.

Solution: Restore the registry settings for the ODBC driver. From the Command Prompts window, execute these commands:

```
regsvr32.exe /s  
  
<UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers  
\UnwiredServer\SQLAnywhere11\bin32\
```

Mobile Business Object Errors

Provides information for problems you may encounter while developing mobile business objects.

Troubleshooting MBOs

Troubleshoot mobile business object (MBO) behavior problems.

MBO retrieves incorrect data

Problem: The MBO does not retrieve the expected data on the device.

Explanation: When you create a relationship to a mobile business object and define a synchronization parameter for the child object, the child object can no longer be used to retrieve all the items. If you want the MBO to retrieve all items, you must create a new MBO without relationships, or create the relationship to the MBO and set the synchronization parameter. Set the synchronization parameter with a default or a personalization key, or allow the user to input the value of the synchronization parameter.

Solution:

1. To retrieve all items, you must create a new MBO.
2. To use the original child mobile business object without a relationship, set the synchronization value on the device. For example, if you create a screen with only the child mobile business object, then you must set the synchronization parameter with personalization. If you do not set personalization, then no data is shown on the device for the client MBO.

Incorrect data for child MBO

Problem: If you update operations on a child MBO and synchronize the parent MBO, the updated data shows correctly on the parent MBO, but a pending status record still shows on the child MBO. This also affects table and list detail MBOs.

Workaround: If the MBO to which you are adding a synchronization action has a child MBO, you must add a Refresh action to the screen of the child MBO so that the screen refreshes with the updated data.

No Data or Images Appear on the Device

Problem: No data or images appear on the mobile device.

Explanation 1: This may indicate that operation parameter datatypes (`int`, `decimal`, `image`, and so on) are mapped incorrectly. You map parameter datatypes to strings from the Mobile Application Diagram when you create an MBO operation.

Solution:

1. From the Mobile Application Diagram, double-click the operation to show in the Properties View.
2. Select the **Parameters tab**.
3. Select a datatype in the first **Datatype** column that matches the one in the second Datatype column. For the image type, you need to provide binary type, and the size of the binary.
4. You may need to configure the specified column, using the Edit Box on Screen Design, to show the contents on the device correctly.
5. You may also need to update the logical type, similar to image type, to show the contents on device correctly.

Explanation 2: The synchronization result may be empty if these rules are not followed:

- When the backend is a SQL Anywhere database, and using the `CHAR NULLABLE` column as a sync parameter, you do not need to pad the value with spaces. When using the `CHAR NOT NULLABLE` column as a sync parameter, you need to pad the value with spaces.
- When the backend is a non-SQL Anywhere database, and using the `CHAR` column as a sync parameter, you always need to pad the value with spaces.

Solution: Re-evaluate the synchronization parameter value, and make modifications using these rules as a guide.

Explanation 3: This may indicate that a new consolidated database in an existing SQL Anywhere environment needs trailing spaces enabled to perform proper comparison by ignoring the trailing blanks. When device client provides a synchronization parameter value, it must include trailing spaces. UTF8 collations are also a requirement for Unwired Platform.

Solution: See *System Administration for Sybase Unwired Platform > Environment Setup > Database Setup > Initializing a New Consolidated Database* for information about setting up the environment properly.

Troubleshooting Improperly Designed Mobile Business Objects

Problem: Improperly designed MBOs can lead to mixed data from different users being returned to the device application.

Each of these designs are improper:

- The MBO includes a synchronization parameter and it is exposed to the client (allow the client to supply values as load parameters), but the synchronization parameter does not map to an attribute.
- You customize the download SQL and do not use the synchronization parameter in the download SQL.

Solution: Avoid these design mistakes.

Database MBO Fails on Execution

Problem: A database mobile business object (MBO) that worked during preview, fails (does not display table data) when it is deployed and executed.

Explanation: This may happen if a database MBO created from an Adaptive Server® Enterprise stored procedure uses chained transaction mode. By default, MBO development preview uses SQL in chained transaction mode, so any changes made to the data source are rolled back to avoid permanently changing data. The result is the MBO behaves as expected when previewed, but fails when a device application attempts to synchronize the MBO. An error is logged in the Unwired Server<hostname>-server.log file.

Solution: Check the <hostname>-server.log file for errors related to chained transaction mode. If you see them, use Sybase Unwired WorkSpace to execute a SQL statement that contains `sp_procxmode <procname>, anymode`, where <procname> is the name of the stored procedure, so the stored procedure runs in autocommit mode.

Web Service MBO Sync Exceptions After an Interface Change

Problem: After changes are made to a Web service interface, you see sync exceptions and related errors in the client log. Server log errors may also indicate that Unwired Server did not recognize the value of HTTP Header SOAPAction: `http://sybase.com/getPersonalKey`.

Explanation: If changes are made to the Web service interface, you must make corresponding changes to the Web Service mobile business object (MBO).

Solution:

1. Change the connection profile for the new WSDL.
2. Re-create the MBO.
3. Redeploy the MBO.

Note: Alternatively, change the connection profile for the new WSDL, and do a refresh to recreate the MBO.

Binding an MBO to a Database Fails

Problem: Binding a mobile business object (MBO) to non-Sybase database fails.

Solution: If you are using a IBM, DB2, Oracle, or Microsoft SQL database, you must place the corresponding JAR file into the `lib` folder in the associated plugin. Before you create or bind an MBO using any of these databases, select the JAR from the appropriate directory when you create the corresponding JDBC. Search *Sybase Unwired WorkSpace - Mobile Business Object Development* for the topic *Configuring Your Environment to Use a JDBC Driver*.

Structure MBOs Not Automatically Generated After Rebinding

Problem: When rebinding an MBO, Unwired WorkSpace keeps the existing parameters, so you must manually create parameters, set correct data type for the parameters, then map the

parameters to the new arguments. But if the argument is a complex (structured) type created during the rebind, Unwired WorkSpace cannot automatically generate the new complex type, and the user cannot set the new complex type to a parameter which causes parameter and argument mapping failure.

Workaround:

1. If rebinding the MBO to the same datasource (for example, rebind an SAP MBO to another SAP BAPI) – edit the definition in the Definition Edit dialog instead of the Rebind wizard.
2. If rebinding the MBO to a different datasource – after rebinding, manually create the needed complex type. Then manually map parameters and arguments in the Properties view.

Multilevel Insert Operation Fails

Problem: A multilevel insert operation fails on the device client.

Explanation: A multilevel insert operation can fail for a variety of reasons, such as the attribute and parameter relationship are not set up properly, or the relationship definition is not correct.

Solution: Investigate how the multilevel insert operation is designed, and make appropriate corrections.

Search Errors and the RTRIM Function

Problem: When performing a search with "equals" and "ends with" search criteria, you see an Unable to find... message.

Problem: When performing a search with "does not end with" search criteria, all the rows are shown.

Solution: If your table includes the char column, you can use SQL with **rtrim(column name)** as the column name in the MBO definition. For example:

```
SELECT dept_id,
       rtrim(dept_name) as dept_name,
       dept_head_id
FROM sampledb.dba.department
```

Update Operation Fails

Problem: An update operation fails if **set maxlength** for the attribute is less than the length in the enterprise information (EIS) database column.

Explanation: If **rtrim** is not used in these cases, the update operation fails with this exception:

```
com.sybase.vader.da.jdbc.MinCheckAffected
RowCountException.null
```

Solution: Use the **rtrim** function in the SQL statement to update operations for **char** datatype columns, such as:

```
SELECT dept_id,
rtrim(dept_name) as dept_name,
dept_head_id FROM sampledb.dba.department
```

Relationship Error

Problem: A mobile business object (MBO) with multiple relationships to a single child MBO reports an error.

You see an error similar to this in Unwired WorkSpace:

Error: The target mobile business object 'MBO_name' can not be used in relationship 'relationship_name' since it already was used in a one-to-many or one-to-one relationship as a target.

A child mobile business object (MBO) with multiple relationships, supports only one cascade relationship.

Solution: Examine all of the relationships to a single child MBO and unselect **Composite** for all but one.

Note: Relationship errors can happen for a variety of reasons. For information about relationship validation rules and development guidelines, see:

- *Troubleshooting > Validation Rules > Unwired WorkSpace Validation Rules and Error Messages > Relationship Validation Rules and Error Messages*
 - *Sybase Unwired WorkSpace - Mobile Business Object Development > Develop > Working with Mobile Business Objects > Mobile Business Object Data Properties > Creating Relationships Between Mobile Business Objects > Relationship Guidelines*
-

SQL Statement Yields Unexpected Results

Problem: A SQL statement yields unexpected results in runtime.

Explanation: This may happen when you use a non-Sybase database, and include a comparison operator in a definition SQL statement followed by a parameter; for example: "select * from ... where columnA=:parameterA, and the default value of this parameter is set to null.

Solution: If you need null as the default value, define an object or operation and manually write the definition SQL statement (for example, " columnA is null") in the Property view.

Warning Message When Creating Relationships

Problem: When attempting to create a Mobile Business Object (MBO) relationship from the palette, a warning message is displayed.

Explanation: A relationship between a normal MBO and a structured MBO is not allowed; and a relationship between a normal MBO and a local business object is not allowed.

Relationships can be created only between normal MBOs, or between local business objects. No cross-over cases are supported.

Solution: Create a relationship between normal MBOs, or between local business objects.

Synchronization Fails if Binary Attributes Have Unspecified Length

Problem: Synchronization failures may occur for attributes with binary data types if no length is specified.

Explanation: When generating object API code for projects that contains an MBO with an attribute whose data type is BINARY, and no length is specified, this warning message displays:

```
Page size is smaller than the maximum
business object size, which may cause
synchronization failure.
```

Solution: You have two options:

- Specify a length for the attribute with BINARY data type if you know, for example BINARY(100).
- Do not map this attribute to a synchronization parameter, or synchronization may fail.

Preview Errors in Unwired WorkSpace

Problem: Preview fails due to transaction errors.

Explanation: By default, Unwired WorkSpace runs in chained transaction mode when you select the preview option for a mobile business object. This lets the transaction roll back after preview so the enterprise information system (EIS) data is not affected.

In some cases, the EIS operation cannot function in chained transaction mode. For example, in these two typical scenarios:

- A stored procedure was created in autocommit mode, and `sp_procxmode anymode`, `<procedure name>` has not been run to allow chained transaction mode.
- When creating a mobile business object for a stored procedure in Adaptive Server® Enterprise 12.x, the **drop temptable** command fails.

Queries that use DDL, for example, a `SELECT . . . INTO #temptable`, that creates a temporary table, and then after processing data in the temp table a `DROP TABLE #temptable` generally cannot be run in chained mode. You receive this error:
Execution failed: The 'DROP TABLE' command is not allowed within a multi-statement transaction in the 'tempdb' database.

Solution:

1. Use a text editor to open %SUP_HOME%\Eclipse\UnwiredWorkSpace.bat.
2. Locate **-vmargs** in the script and add the following after the argument (with a space between the argument and this entry):
`-DautocommitPreviewTransaction=true`
3. Save and close the file.
4. Restart your workspace.

Preview Errors for a Database Mobile Business Object

Problem: You see an error when you are previewing a database mobile business object (MBO) that uses stored procedures.

```
Execution failed: Stored procedure 'pubs2.dbo.byroyalty' may be run only in unchained transaction mode.
```

During preview, the connection is set to chained mode so the query can be rolled back. **set chained off** causes the current session to use unchained transaction.

Solution: Issue `sp_procxmode 'pubs2.dbo.byroyalty', anymode.`

Preview Errors for a Web Service Mobile Business Object

Problem: You see an error when you are previewing a Web service mobile business object (MBO).

Explanation: If the Web service **datetime** datatype includes time zone information, the preview data may not be correct. This issue stems from the different time zones between the enterprise information system (EIS) and the device client.

Solution: See *Sybase Unwired WorkSpace - Mobile Business Object Development* for information about time zone datatype behavior, and modify the Web service MBO accordingly.

Compiler Errors When Deploying Device Applications

Problem: You see compiler errors when you deploy mobile business objects (MBOs) that begin with the same characters and end with "list."

If there are two MBOs named *xxx* and *xxxList*, for example "OpGet" and "OpGetList" in the same project, compiler errors occur when the device application is deployed to either Windows Mobile or BlackBerry devices. Internally, the compiler generates a List class for MBOs. In the case of OpGet, an entity named OpGetList is generated, which conflicts with the second MBO named OpGetList, resulting in compiler errors.

Solution: Avoid this naming convention if it conflicts with other MBOs. Rename existing MBOs or place them in a different project.

Deploying to Unwired Server Fails

Problem: Deployment fails with when you deploy a mobile business object (MBO) to Unwired Server.

Explanation: This may happen for a variety of reasons, for example, if both attribute mapping and parameter mapping are identical in an MBO relationship; or if the data sources bound to MBOs are not up and running, the Deployment wizard will fail.

Solution 1: Investigate the MBO to make sure it is configured correctly, then make corrections. For the example of identical mappings, avoid identical mapping in attributes and parameters that have a relationship.

Solution 2: Make sure the data sources associated with the MBOs are up and running.

Deploying to Unwired Server in a Cluster Fails

Problem: Deployment fails when you deploy a mobile business object (MBO) to Unwired Server in a cluster.

Explanation: When you deploy a package into a Sybase Unwired Platform cluster, you must be connected to the Unwired Server designated as the primary server in the cluster. If not, deployment fails and you see a message similar to following in the server logs:

```
YYYY-MM-DD HH:MM:SS.SSS WARN      MMS      Thread-466
[com.sybase.sup.server.SUPDomain] On Non primary servers, you could
not
execute createDsyncDataZip()
java.lang.Exception: On Non primary servers, you could not execute
createDsyncDataZip()
    at
com.sybase.sup.server.cluster.ClusterUtils.createDsyncDataZip
(ClusterUtils.java:816)
    at com.sybase.sup.server.SUPDomain.incrementDomainVersion
(SUPDomain.java:779)
    at com.sybase.sup.admin.server.DomainAdminImpl.deployJar
(DomainAdminImpl.java:291)
```

Solution: Set up a connection profile to the server that has become the primary, and deploy using that connection. See *Sybase Unwired Workspace - Mobile Business Object Development > Configure > Configure - Eclipse Development Environment > Creating a Sybase Unwired Server Connection Profile* for how to create a new Sybase Unwired Server connection profile.

Modified Port Changes Not Deployed

Problem: Modified data source port information is not reflected when you deploy mobile business objects (MBOs).

For example:

1. Create a data source named DB1 that uses port 4100.
2. Create an MBO that references this data source.
3. Deploy the MBO to Unwired Server.
4. In Enterprise Explorer, change port information for the connection profile to 41007.
5. Redeploy the MBO to Unwired Server.

During the second deployment, the endpoint port remains the original value, 4100, rather than 41007.

Solution 1:

1. Create an XML configuration file that contains the information to be modified, as this `chgEndpoint.xml` file illustrates:

```
<?xml version="1.0" ?>
<project name="DDSOPS1" default="configure">

  <import file="ant-config-tasks.xml" />
  <import file="default-properties.xml" />
  <target name="configure">
    <setProperties dataSource="DDSOPS1" merge="false">
      <property name="databaseType" value="Sybase_ASE" />
      <property name="user" value="sup_cv" />
      <property name="password" value="xyz" />
      <property name="portNumber" value="41007" />
      <property name="serverName" value="ddsops1" />
      <property name="databaseName" value="tsbde" />
    </setProperties>
  </target>
</project>
```

2. Save the XML file in the `<UnwiredPlatform-InstallDir>\UnwiredPlatform\Servers\UnwiredServer\config` directory.
3. Run **configure.bat** using the XML file as an argument from the command line in `<UnwiredPlatform-InstallDir>\UnwiredPlatform\Servers\UnwiredServer\bin`. For example:
`Configure chgEndpoint.xml`

The port information is updated and reflected during deployment.

Note: The basic XML Ant scripts are the same for all data source types. The properties vary depending on endpoint types.

For each JDBC data source, the database vendor's specific JDBC driver may expose its own connection properties. You can specify those properties in an XML configuration file by prefixing the property name with "jdbc:".

For example, the Sybase jConnect™ for JDBC™ driver contains an IS_CLOSED_TEST connection property that you can set. This property can be a SQL query that jConnect sends to the database when the application calls Connection.isClosed(), or "select 1". If the query successfully executes, the connection remains open, and returns false. The jConnect driver also has an "INTERNAL" setting that enables jConnect to remember the state of the connection the last time it was used. This setting prevents the database from being pinged multiple times, especially in an appServer context where the isClosed() check is performed each time a connection is checked into or out of a connection pool. To set the "INTERNAL" property from the Ant script, add:

```
<property name="jdbc:IS_CLOSED_TEST" value="INTERNAL" />
```

See your JDBC driver documentation for information about driver specific connection properties.

Solution 2:

1. In the Unwired Server installation directory, change to this folder:
`<UnwiredPlatform-InstallDir>\UnwiredPlatform\Servers
\UnwiredServer\Repository\Instance\com\sybase\djc\sql
\DataSource\`
2. Open the `.properties` file and change the port number.
3. Redeploy the MBO package from Sybase Unwired WorkSpace, using the same Unwired Server connection when prompted in the deployment wizard.

Troubleshoot Code Generation

Provides troubleshooting topics for generating device application code using the client object API.

SQL_MAX_ROW_SIZE_EXCEEDED

Problem: You encounter a SQL_MAX_ROW_SIZE_EXCEEDED error during client object API code generation. This means the validation process has detected one or more size problems in your mobile business object (MBO).

Solution: Review your MBO design, and make the appropriate modifications using this information. Some possibilities:

- Set the maximum length of string attributes to a smaller number (the default value is 300).
- Change the maximum length of string attributes to 8192. This changes the SQL datatype of the attribute to "LONG VARCHAR".
- Redesign the MBO by removing some attributes.
- Use the proposed page size in the code generation wizard.

Error When Running Generated 32-bit Windows Code on a 64-bit Machine

Problem: An error occurs in the client API code when myDatabase initializes on line :"`_pid = com.sybase.afx.util.PidUtil.GetId();`".

Solution: If you encounter an error when running generated 32-bit Windows device client code on a 64-bit machine:

1. In Visual Studio, in the toolbar, select **Configuration Manager** from the Solution Platforms list.
2. In the Configuration Manager, in Active Solution Platform, select **New**.
3. In **Type or select the new platform**, enter x86.
4. Restart the application.

Namespace Name Does Not Exist Error

Problem: When generating C# code, this error appears: The type or namespace name <'name'> does not exist in the namespace 'SUP.name.com (are you missing an assembly reference?)'.

The full message, including an example, appears as:

```
public sealed class SalesforceAutomationDB :
com.sybase.afx.DatabaseBase
{
....
}

Error 1 The type or namespace name 'sybase' does not exist in
the namespace 'SUP.Mobile.com' (are you missing an assembly
reference?)
C:\Documents and Settings\username\My Documents\Visual Studio
2008\Projects\SFA\SFA\Generated Code\src\SUP\Mobile\com
\SalesForceAutomationDB.cs 41 54 SFA
```

Explanation: The namespace name does not follow C# namespace conventions. In this example, avoid using patterns that include com, and embedding a name, such as Sybase:

- (xxx.)+com(.xxx)*
- (xxx.)+Sybase(.xxx)*

Solution: Change the namespace name to follow C# namespace conventions, and regenerate C# code. In this example, the conventions would not use com, and would include the Sybase name in the format Sybase.Persistence.

Windows Mobile Code Generation Fails During Compilation

Problem: The code generation process failed. The Unwired Server log indicates an error similar to: MSBUILD : error MSB1009: Project file does not exist.

Explanation: The default character encoding of Windows 7 is not consistent with the Java default String encoding. For example, if the generate project target path contains non-ASCII characters, the compilation fails.

Solution:

- Only use the ASCII characters in the path and solution name.
- You can also open the generated project in Microsoft Visual Studio and compile, and deploy the project to the device.

Cannot Generate Code with Javadoc Option Enabled

Problem: When using the code generation wizard, with the option enabled to generate javadoc, you cannot generate the code to a folder.

Explanation: This typically occurs if the folder name contains Chinese characters.

Solution: To generate the code to a folder containing Chinese characters, you must first change the Language for non-unicode programs to Chinese.

1. From the Start menu, select **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.

Code Generation Fails

Problem: Code generation fails or takes a very long time.

Explanation: This may indicate that the data sources bound to mobile business objects (MBOs) are not up and running, which prevents the binding to take place. The resulting performance impact causes code generation to fail.

Solution: Make sure the data sources associated with the MBOs are up and running.

Troubleshoot Custom Development

Provides custom coding device applications using the client object API.

Insert Operation Fails for iOS

Problem: A mobile business object (MBO) insert operation fails with a runtime exception for iOS; however, the same MBO works on Windows Mobile.

Solution: For an attribute that cannot be null, assign a value if the corresponding Property datatype in the generated MBO Objective-C code is a class (pointer).

When developing an iOS client:

- If a non-nullable attribute datatype is converted to a nonprimitive datatype (for example, class `NSNumber` or `NSDate`) in the iOS client application, verify that the corresponding property for the MBO instance is assigned a value; otherwise the application may issue a runtime exception when creating a new MBO instance.
- If an Adaptive Server Enterprise column datatype is numeric, during the MBO creation, the datatype of the corresponding attribute is decimal and non-nullable. In generated Objective-C MBO code, the corresponding datatype is `NSNumber`. When creating a new MBO instance, assign this property a value, otherwise the application may issue a runtime exception (its actual value is later replaced by Unwired Server).

MBO Update Fails

Problem: A mobile business object (MBO) that you edited using the C# client API fails to update upon synchronization.

Solution:

1. Check the Logs Screen for the detailed log record.
2. If the log record does not provide enough information, check the C# client API code for errors, for example:

```
int id = 777; //it's the customer id
Customer cust = Customer.FindBy(id);
cust.Name = "some other name";
cust.Update();
cust.SubmitPending();
DBClass.Synchronize();
//...
//you verify operation log that indicates the update failed at
backend server
//...
```

```
cust = Customer.FindBy(id);
cust.Delete();
cust.SubmitPending();
DBClass.Synchronize();
//...
//you verify operation log that indicates the delete failed
//...
```

Call **CancelPending()** after the failed update operation to delete the MBO, for example:

```
int id = 777; //it's the customer id
Customer cust = Customer.FindBy(id);
cust.Name = "some other name";
cust.Update();
cust.SubmitPending();
DBClass.Synchronize();
//...
//you verify operation log that indicates the update failed at
backend
servercust = Customer.FindBy(id);
//...
Customer customer = Customer.FindById(id);
if(errorHappened)
{
    Customer.CancelPending();
}
else
{
    customer.SubmitPending();
}
//...
//you verify operation log that indicates the delete succeeded
//...
```

Windows Mobile Compile Error For Generated API Project

Problem: A Windows Mobile compile error is encountered when generating a client object API project.

For example: Error 6 'entity.<variable>' does not contain a definition for '<variable>' and no extension method '<variable>' accepting a first argument of type 'entity.<variable>' could be found (are you missing a using directive or an assembly reference?) C:\work\entity_Sol\entity_MBO\entity\intrnl\Department_rs.cs 45 46 entity_MBO

Explanation: This error is due to variable name conflicts. The package name includes client object API reserved words (in the example, "User" is used by OfflineAuthentication.cs, and "entity" is the field of an internal operation). The compile process could not resolve package name with the internally reserved words, and reported the error.

Solution: Rename the package, then rerun the compile.

Cannot Insert DEFAULT in Timestamp Column

Problem: Cannot insert DEFAULT in the timestamp column using the Object API.

Explanation: DEFAULT is a Microsoft SQL keyword, and cannot be set by the client.

Solution: Use this create operation instead:

```
insert into sql_date (c_datetime, c_smalldt) values (:dt1, :dt2)
```

Synchronization to SAP Data Source Repeats Excessively

Problem: Created an MBO that is calling a customized RFC with an input parameter. The first synchronization to the SAP data source works correctly. The second sync executes the first RFC, then a second one. This causes failures in SAP, because the new update cannot be transmitted.

Client Names

Explanation 1: Synchronization parameter crosstalk between clients could be causing a problem if multiple clients are running concurrently using the same name. Synchronization parameters are stored on Unwired Server by user name.

Solution: Avoid the issue by running each client with a different user name. It is also necessary to pass the username/password combination using `LoginToSync`.

MBO Read Operations

Explanation 2: If you map your MBO's read operation (through which you retrieve the MBO attributes) to a backend operation that commits a BAPI, then keep in mind that each time you perform a synchronization, the sync parameter is stored on Unwired Server. Later, when you sync it again with another sync parameter value, all the stored sync parameter values are synced again.

Solution: To avoid this problem, call `SyncParameter.delete()` to clear the cache after each synchronization, for example:

```
<name>SyncParameter param =
nameMBO.getSynchronizationParameter();
param.delete();
param = nameDB.getSynchronizationParameter(); //must re-get the sync
parameter instance
param.set... //set new sync parameter value
param.save();
```


Troubleshoot Mobile Workflow Forms Editor

Provides troubleshooting information about using Mobile Workflow Forms Editor to design, develop, and test device workflows, and to generate code.

See the *Validation Rules* section for information about validation rules and error messages.

These log files are useful for troubleshooting:

- WorkflowClient – shows the interaction between the Mobile Workflow and the data sent to and from the Unwired Server (to enable, turn workflow client tracing on).
- Sybase Messaging Client (located on the mobile device; not available on iOS) – shows the interaction between the Sybase Messaging Client on the mobile device and Unwired Server. On a mobile device that has the Sybase Messaging Client installed, view **Sybase Settings > Show Log**.
- Unwired Server – shows activity occurring on Unwired Server.

Operation Fails with NULL Values Not Supported Error

Problem: The operation fails, and `NULL <datatype>` values are not currently supported in the Sybase Message Client appears in the WorkflowClient log.

Explanation: The Sybase messaging client cannot differentiate between a NULL value and an empty string. On the device client, there is no way for the user to enter a NULL value for a particular field, so the operation fails.

Solution: Modify the mobile workflow to not use NULL values

Operation Fails with MinCheckAffectedRowCountException Error

Problem: An operation fails, and `com.sybase.vader.da.jdbc.MinCheckAffectedRowCountException` appears in the WorkflowClient log.

Explanation: The client is specifying outdated values for an update operation, which do not match the values in the back-end data source.

1. In the workflow client log, if you see `com.sybase.vader.da.jdbc.MinCheckAffectedRowCountException`, check the preceding line, which should include `JsonContent`.

This line contains the old values that are being sent from the client.

2. Check these values against the outdated values in the back-end data source, determine which are different, and make appropriate changes.

Mobile Workflow Is Not in the Client List

Problem: The mobile workflow does not appear on the client device.

Explanation 1: The mobile workflow has not been deployed, or has been deployed, but not assigned to the device.

Solution:

1. Ensure that the user's device has the Sybase messaging client installed, and that it is configured correctly.
2. Use Sybase Control Center (**Device Users > Devices**) to ensure that the device is connected.
3. Deploy the mobile workflow, and assign it to the user's device.

Explanation 2: The mobile workflow does not have a starting point.

Solution: Add a starting point to the mobile workflow.

Explanation 3: The device cannot connect to Unwired Server.

Solution:

1. Check the Sybase messaging client log file for the error `Cannot Connect To Server (503)`.
2. Ensure the settings on the mobile device (**Sybase Settings > Connection**) are correct.
3. Ensure that Unwired Server is running, and that the mobile device has connectivity to Unwired Server.
4. If you are using a Windows Mobile emulator, ensure that Active Sync (Windows XP) or Windows Mobile Device Center (Windows Vista, Windows 7) is running. In the Device Emulator Manager, uncradle and recradle.

Explanation 4: The user attempted to set up a previously connected user on another mobile device, or on a mobile device emulator that has been reset.

Solution:

1. Check the Sybase Messaging Client log file for the error `Wrong Device For Code (579)`.
2. Use Sybase Control Center (**Device Users > Devices**) to delete the user and reregister the user.

Mobile Workflow Does Not Receive E-mail Messages

Problem: A mobile workflow, which includes an e-mail starting point, does not receive incoming e-mail messages.

Explanation 1: The workflow has not been deployed, or has been deployed but not assigned to the device

Solution:

1. Ensure that the user's device has the Sybase Messaging Client installed, and that it is configured correctly.
2. Use Sybase Control Center (**Device User > Devices**) to ensure that the device is connected and online.
3. Deploy the mobile workflow, and assign it to the user's device.

Explanation 2: The e-mail server is not started, or is not functioning properly.

Solution:

1. Restart the e-mail server if necessary.
2. Send a test e-mail message to the device using the Send E-mail option on the Flow Design page of the Mobile Workflow Forms Editor.

Explanation 3: The matching or e-mail transformation rules are improperly configured, or the e-mail message is improperly formatted.

Solution:

1. Verify that the matching rule and e-mail transformation rules are correct and conform to the e-mail being sent to the device.
2. Send a test e-mail message to the device using the "Send a notification..." option on the Flow Design page of the Mobile Workflow Forms Editor.

Explanation 4: The Notification Mailbox is not properly configured.

Solution:

1. Use Sybase Control Center to verify that the Notification Mailbox is properly configured.
2. Navigate to **Workflows > Notification Mailbox** and ensure that the Enable checkbox is selected.
3. Ensure there is a distribution rule (for example, the To: field must contain the value Custom 1) that matches the properties for the device user (Frank's iOS device's Custom 1 property is set to frank@localhost).

Object Query Listview Does Not Display Data

Problem: A listview that is populated from an object query does not display any data.

Explanation: The listview key does not match the key being returned from the object query.

Solution:

1. Ensure the appropriate key is selected by clicking the drop-down box for the listview's Key field
2. Redeploy the workflow.

Operation Fails with Package not Found Error

Problem: The operation fails, and `Package not found` appears in the workflow client log.

Explanation: One or more mobile business objects (MBOs) has not been deployed to Unwired Server.

Solution: Deploy MBOs and retry the operation.

Operation Fails with Login Failed Error

Problem: The operation fails, and `loginFailed:true` appears in the workflow client log.

Explanation: The user name not may not be set properly.

Solution: Make sure that the correct user name and domain (for example, `supAdmin@default`) are specified. Look for `username: supAdmin@default` in the workflow client log.

Sybase Settings Program Does Not Appear on Windows Mobile Device

Problem: The Sybase Settings program does not appear on the Windows Mobile device, after installing the Sybase messaging client to an actual Windows Mobile device with an ActiveSync connection.

Explanation: This may happen if you installed silently to the device. Install with messages turned on.

Solution: Unselect the Silent Install option, then redeploy the Sybase messaging client.

Failed to Create Second Process

Problem: While trying to view a mobile workflow application on the BlackBerry simulator, the operation fails and a message like Failed to Create Second Process appears.

Explanation: This may happen if you attempt to open more than one webkit for the BlackBerry simulator. The BlackBerry simulator has a known limitation of only allowing one webkit instance to run. If you have already opened the browser, then try to open a workflow, you will receive an error like Failed to Create Second Process.

Solution:

1. Close the other webkits.
2. Reopen the workflow application.

Customized Code is Blocked

Problem: The listview control is blocking a customized section of the `workflow.html` so the customized section is not loaded.

Solution:

1. Open the generated `workflow.html` file, located in the Generated Workflow \html folder, with a text editor.
2. Set the data-role attribute to "content." For example:

```
<div class="customTopOfFormStyle" data-role="content" id="topOfStart_ScreenForm"></div>
```
3. Save and close the `workflow.html` file.
4. In the Generate Mobile Workflow Package dialog, ensure that you uncheck the Generate option, and redeploy the Mobile Workflow application to Unwired Server.

Note: The `workflow.html` file is overwritten every time you regenerate the mobile workflow package.

Troubleshoot Device Clients

Provides troubleshooting information for device client applications that are typically deployed in the field.

Device Provisioning

Provides troubleshooting information for provisioning device application clients using Afaria.

Device Cannot Communicate with Afaria Server

Problem: When Afaria is part of the same mobile enterprise as Unwired Platform, Afaria Server appears to not respond to incoming requests from Unwired Platform components.

Solution:

- Ensure that Afaria Server is running.
- Check whether the device browser can open Web pages correctly to determine whether device network access is functioning.

Device Clients

Provides troubleshooting information for device application clients. Some topics may apply to device application clients in general, while others may be specific to platform types.

Synchronization Fails in a Remote Server Environment

Problem: Synchronization fails due to an error on the server.

Explanation: This may happen if you configured `sampledb` in a remote server environment, and accepted the default server name "localhost." The default server name causes synchronization to fail.

Solution: Ensure that the `sampledb` server name reflects the exact IP number or server name of the remote server:

1. Launch Sybase Control Center, and, in the Administration perspective, open the remote cluster.
2. In the left navigation pane, expand the appropriate domain, and click **Connections**.
3. In the **Connections** tab, select **sampledb**, and click **Properties**.

4. In the Properties dialog, edit the sampledb server name value to reflect the appropriate remote server name or IP address.

Synchronization Fails With OutOfMemory Error

Problem: The device application seems to hang for 15 or 20 minutes, then synchronization fails with a `java.lang.OutOfMemoryError` or `OutOfMemoryError` message.

Explanation 1: The load and synchronization parameters may not be set correctly to eliminate extraneous data. The volume of synchronized data, both from the enterprise information system (EIS) to the consolidated database, and from the consolidated database to the device client is causing the out-of-memory error.

Solution: Redesign the mobile business object (MBO) to take better advantage of both the load and synchronization parameters. Search *Sybase Unwired WorkSpace - Mobile Business Object Development* for the topic *Combining Load and Synchronization Parameters* to learn more.

Explanation 2: The device database page size may be too small.

Solution 1: Increase the page size for the device, using the Sybase Unwired WorkSpace code generation wizard. Search *Sybase Unwired WorkSpace - Mobile Business Object Development* for the topic *Generating Object API Code*, or search the appropriate *Developer Guide* for information about generating code for a specific device platform.

Timeout Errors During Client Synchronization

Problem: Various timeout errors are reported during client synchronization, especially in a cluster configuration.

Explanation: This may indicate that performance tuning is required to make Unwired Server processing more efficient.

Solution: Take one or more of the following actions to improve performance.

Action 1: Review the Configuration

Review the cluster configuration for settings that may impact performance.

1. Monitor the CPU utilization during client synchronization to determine the percent used. Utilization should be less than 80% for the cluster; ideally no more than 70 – 80%. If it is higher, check the Unwired Server logs for related messages.
2. High CPU utilization may indicate the cluster is bound by physical I/O (the disk queue is over utilized). The administrator may check the performance monitor to confirm this state. When the disk queue is backed up, requests may start to time out and the overall system throughput will degrade. Follow the *Sybase Unwired Platform Performance and Tuning* white paper instructions to reduce the load on the cluster by decreasing the number of MobiLink threads on each SUP node. The white paper is located on the SAP Developer Network Web site: <http://www.sdn.sap.com/irj/sdn/mobile>. Click on Sybase Unwired Platform, and navigate to the white paper category.

Action 2: Reconfigure Client Connections

Configure the client connections to use a longer timeout on the connection. The default is the minimum value of 240 seconds, and the maximum value is 600 seconds. This value is set by adding the timeout property and desired value to the `NetworkStreamParams` on the connection.

For CS/UltraLite, assuming the generated DB class name is "MyPackageDB", the code to set timeout could be:

```
MyPackageDB.GetSynchronizationProfile().NetworkStreamParams="trusted_certificates=;url_suffix=;timeout=600"
```

For Java/UltraLiteJ, you cannot set timeout, because UltraLiteJ does not provide an API to set a timeout value for client connection. Sybase Unwired Platform generated client code for UltraLiteJ (such as for Java desktop and RIM replication based synchronization client) only supports the "trusted_certificates" and "url_suffix" properties in `NetworkStreamParams`. If you set other properties in `NetworkStreamParam`, they will be ignored.

For additional information on setting the `NetworkStreamParams` properties, search *System Administration* for references to "stream parameter." The topic *Verifying Device Client HTTPS Setup* provides this sample code for setting stream parameter:

```
"url_suffix=/ias_relay_server/client/rs_client.dll/[SUP_FARM_ID];tls_type=RSA;trusted_certificates=rsa_root.crt;identity=id_client.pem;identity_password=pwd;"
```

Action 3: Modify Client Application Code

Add retry logic to your client application code to retry syncs that fail with timeout.

Device Operation Fails With SQLE_NOT_IMPLEMENTED Error

Problem: A Windows Mobile or BlackBerry device throws an Exception with the message `SQLE_NOT_IMPLEMENTED`.

Solution: If the device client database throws the exception `SQLE_NOT_IMPLEMENTED` in a JOIN query, check your join fields to make sure they are the same type.

No Data Displays on Device Client

Problem: No data displays on an iOS or Windows Mobile device, and no error message displays.

Explanation: This may happen because of a subscription failure. Because of the failure, no data is downloaded to the device. Possible causes:

- iOS – the DOE Connector Listener URL is not correct.
- Windows Mobile - the client is not connected to the messaging service (the user displays as offline in the Sybase Control Center administrative console). The DOE Connector ESDMA package did not deploy properly (especially if the Listener URL is not correct).

Solution:

1. Verify that the DOE Connector Listener URL is correct.
2. Redeploy the DOE Connector ESDMA package.

See:

- *Sybase SAP DOE Connector Installation Guide*
- *System Administration for Sybase Unwired Platform > System Reference > EIS Data Source Connection Properties Reference > SAP DOE-C Properties*
- *Sybase Control Center > Configure > Configuring Unwired Platform > Packages > Configuring Messaging and DOE-C Package Device Settings*

Synchronizing the Device Application Returns Only One Row of Data

Problem: If the enterprise back-end database table has no primary key and contains multiple rows of identical information in every column, a device application user can synchronize only one row.

Explanation: If a device application user tries to create multiple rows that are identical in every column, it succeeds to update the back-end database, but only displays one column on the client and the consolidated database (CDB).

Solution: verify the back-end database has a primary key.

Packet Dropped State

Problem: User stops receiving packets for a package subscription and "Packet Dropped" displays as the subscription status for the user and device ID.

Explanation: A problem has occurred in processing the packet. No further packets sent to subscriptions for this package will be processed until the problem is fixed.

Solution:

1. In the Unwired Server log, locate the lines recording the packet drop.
 - a. In a text editor, open `<SUP_installdir>\Servers\UnwiredServer\logs\<server_name>-server.log`.
 - b. Search for "packet-dropped" or "packet dropped" – you should find two consecutive lines that show this text far to the right:

```
... Detected a packet-dropped candidate message ...  
... DOE-C is negotiating a DOE packed dropped message:
```

2. Determine the reason that the packet was dropped.
 - a. If you see this line below following close after the two lines above, the packet was dropped because the packet drop size was exceeded:

```
MESSAGE: Request Entity Too Large
```

Look further to the right in the first line above and note the "size" value and the "maximum" value:

```
... candidate message - size ####, maximum ####
```

- b. If you see this line below following close after the two lines above, the packet was dropped because the packet read timed out – the packet could not be processed in the time allotted:

Read timed out : Read timed out

- c. If you see neither of the lines above following close after the first two, the packet was probably dropped because it could not be parsed correctly.

3. Use the table below to finish resolving the issue.

Reason for drop	Next action	Resolution options
<p>Packet drop size exceeded</p>	<p>Check with your SAP administrator to see if the packet "size" value can be made smaller than the "maximum" value (the packet drop size).</p>	<p>If your SAP administrator can reduce the packet size below the packet drop size:</p> <ol style="list-style-type: none"> 1. Have the SAP administrator can reduce the packet size. 2. Unsubscribe and resubscribe all affected user devices. <p>If your SAP administrator cannot reduce the packet size below the packet drop size:</p> <ol style="list-style-type: none"> 1. Verify that user devices can handle a packet as large as the one that was dropped. 2. In Sybase Control Center, increase the packet drop size to accommodate the size of the dropped packet: select the package, go to the Subscriptions tab, click the Properties button.
<p>Packet read timed out</p>	<p>Contact your network administrator.</p>	<p>Have your network administrator verify that the timeout was not caused by an unrelated network problem. For example, if the DOE server was unable to resolve the hostname of the Sybase Unwired Server, or a network router failed, a read timeout would result.</p> <p>If you have ruled out network problems as the cause of the timeout, have your SAP administrator increase the packet read timeout value for the package on the SAP end.</p> <p>Then, in Sybase Control Center, increase the packet read timeout value for the package: under Domains, select default, Connections, select the package, then click Properties – DOE SOAP Timeout is the packet read timeout setting.</p>

Reason for drop	Next action	Resolution options
Packet could not be parsed correctly	Contact your SAP administrator.	Have your SAP administrator fix the parsing issue for the package on the SAP end.

For more information on:

- **Understanding packet drop size** – see *System Administration for Sybase Unwired Platform > System Reference > EIS Data Source Connection Properties Reference > SAP DOE-C Properties*
- **Setting packet drop size and read timeout in Sybase Control Center** – see *Sybase Control Center Online Help > Configure > Packages > Viewing and Changing Messaging and DOE-C Package Connection Properties*
- **Setting packet drop size and read timeout with the Command Line Utility** – see *Installation Guide for Sybase SAP DOE Connector > SAP DOE Connector Command Line Utility > Package Management Commands > setEndpointProperties Command.*

Pattern is Too Long Exception

Problem: An exception—The pattern is too long—is reported while trying to update an existing value on a back-end database, via mobile device.

Explanation: This case may happen when a mobile business object's (MBO's) operation definition is generated by dragging and dropping a database table to the object diagram in Unwired WorkSpace.

The error is caused by an incorrect LIKE clause. The LIKE clause should not be used by a text data type whose length is greater than 254 characters, or a binary data type. The LIKE clause is the Unwired WorkSpace's best effort to resolve operations executed against tables that do not contain a primary key.

Solution: Redesign the back-end database to contain a primary key on the offending table.

Feature Not Implemented Exception

Problem: An INSERT operation on the client device results in an exception similar to: Feature 'n/a' not implemented.

Explanation: This may indicate that page size is not large enough to accommodate all MBO attributes included in the mobile application project.

The page size should be larger than the sum of all attribute lengths for any MBO which is included with all the MBOs selected, and must be valid for the database. If the page size is changed, but does not meet these guidelines, object queries that use string or binary attributes with a WHERE clause may fail.

Solution: Regenerate the code, this time specifying a larger page size. For information search *Sybase Unwired WorkSpace - Mobile Business Object Development* for the topic *Generating*

Object API Code, or search the appropriate *Developer Guide* for information about generating code for a specific device platform.

BlackBerry

Provides troubleshooting information specifically for BlackBerry device application clients.

Uncaught exception:java.lang.Error

Problem: You receive an uncaught `exception:java.lang.Error` when starting a device application on a BlackBerry device.

Solution: Verify that the mobile application package containing the mobile business objects referenced by the device application has been deployed to Unwired Server.

BlackBerry Device Login Error

Problem: The BlackBerry device reports login errors for various situations.

Explanation: The login error may be reported for various situations, in addition to login problems.

Solution: Review the Unwired Server log to get the exact error.

BlackBerry Device Synch Fails

Problem: BlackBerry device synchronization fails, and the Unwired Server log reports a `Cannot synchronize` message.

```
Thread-141 [com.sybase.ml.sup.Logger] [-10013] Version  
'default_flightmodelsample.1_0'  
not found in the ml_script_version table.  Cannot synchronize
```

Explanation: A package type mismatch can cause synchronization errors; for example, if the device synchronizes an RBS package with a deployed MBS package, the synchronization will fail.

Solution: Redeploy so the package types match.

BlackBerry Device Client Cannot Sync

Problem: The BlackBerry device client cannot synchronize and reports a `SynchronozationException` message that indicates a `SOLE_COMMUNICATIONS_ERROR`.

Explanation: For applications that use replication-based synchronization (RBS), this may indicate the Unwired Server is down.

Solution: Report that Unwired Server is down, or wait for it to come back online.

BlackBerry Device Cannot Find Valid Certification Path

Problem: The BlackBerry device client cannot set up a secure connection and reports a `SynchronizationException` message that indicates a certification problem.

```
Communication error sun.security.validator.ValidatorException: PKIX
path building
failed: sun.security.provider.certpath.SunCertPathBuilderException:
unable to find valid
certification path to requested targetDetails:
```

Explanation: One of the most common causes for not being able to set up a secure connection is an invalid certificate.

Solution: Make sure the certificate is valid, and the secure connection is set up correctly.

Value INTEGER Out of Range For Destination Error

Problem: When you enter a value on a BlackBerry device in an attribute of the `BYTE` type, you receive this error when submitting the data.

Explanation: This error can happen if the mobile business object is associated with a SQL Anywhere database. With SQL Anywhere, the `tinyint` column is automatically mapped to be an unsigned `TINYINT` type attribute, which can hold values from 0 to 255. However, if you manually map the `tinyint` column to a `BYTE` attribute, on BlackBerry devices, only values from 0 to 127 can be used.

Zero Value Returned Instead of Personalization Key Value

Problem: A zero value was returned on a BlackBerry device instead of the expected personalization key value.

Explanation: This is expected behavior if a default value is not set for a non-nullable personalization key. When a default value is not set for a non-nullable numeric personalization key, the object API can only return a zero for `getXXX` methods.

Solution: If a default value is not set, the device user must set the personalization key value manually. Alternatively, redesign the application to include the default value.

OutOfMemory Error on BlackBerry Simulator

Problem: Received an `OutOfMemory` error on the BlackBerry simulator after inserting a graphic image, saving, and `submitPending` in the client.

Explanation: Assuming the mobile business object (MBO) is set properly with a single binary attribute, this can mean the size of the binary attribute is not sufficient. You may need to expand its size.

Solution: Reset the simulator default heap size to a larger value. You may need to experiment with the BlackBerry simulator you are using. For example, you might develop guidelines such as these after experimenting with a BlackBerry 9550 simulator:

- If you set the simulator heap size to 100MB – the suggested size of the binary attribute is 2.5MB. Beyond that size you will probably see an OutOfMemory error in the client.
- If you set the simulator heap size to 256MB – the suggested size of the binary attribute is 6MB. Beyond that size you will probably see an OutOfMemory error in the client.

If an MBO has more than two binary attributes, Sybase recommends that their combined size not exceed the limit. The OutOfMemory error might not occur if you exceed the limit, but it is a suggested guideline to prevent exceeding the limit.

Server Response Timeout Error on BlackBerry Simulator

Problem: After restarting the BlackBerry client simulator, received a Server Response Timeout error in the event status log (Sybase Setting > Show Log).

Explanation: This typically indicates a problem within the BlackBerry MDS simulator environment. For example, the timeout setting in the BES/MDS configuration may be set too low for Unwired Server.

Solution: Restart the MDS and device simulator frequently to start fresh.

Server Response Timeout Error on BlackBerry Device

Problem: The BlackBerry device client shows a Server Response Timeout error in the event status log (Sybase Setting > Show Log).

Explanation: In general, a "Server Response Timeout" error means that the server did not respond to an HTTP request in a timely manner. This typically indicates that the request was too large or the network too slow to complete a request before either the BES or the network layer on the client timed it out. This could be caused by the BES/MDS configuration being set to a timeout that is too short for the kind of requests being sent (for example, very large uploads or downloads). Additionally, a slowly responding Unwired Server (perhaps because it is overloaded) could cause the same error.

Solution: Check Unwired Server, the network, and BES/MDS configuration settings, and tune for better performance. See the *Sybase Unwired Platform Performance and Tuning* white paper for performance tuning information. The white paper is located on the SAP Developer Network Web site: <http://www.sdn.sap.com/irj/sdn/mobile>. Click on Sybase Unwired Platform, and navigate to the white paper category.

iOS

Provides troubleshooting information specifically for iOS device application clients.

iOS Device Synchronization Fails

Problem: iOS device synchronization fails, and the Unwired Server log reports a `subscribeFailure` message.

Explanation: A package type mismatch can cause synchronization errors. The `subscribeFailure` message imports a log record to the client with the reason for the

failure (such as package not found, or invalid credentials). You can use the database class method **getLogRecords** to get the record that was imported.

Solution: Use message information to solve the problem. For example, you may need to provide valid credentials, correct settings (for example, only MBS is supported for iOS), or reinstall the package on Unwired Server.

iOS Device Cannot Sync

Problem: The device client cannot synchronize with Unwired Server, with errors.

Explanation 1: The Unwired Server may be down. The error would report an `NSEException` that includes `name: loginToSync`, and `reason: No connection to server because the Unwired Server is down or unreachable`. The `loginToSync` method can detect if the server is unreachable (the messaging layer is not in "Connected" state) within a timeout.

Note: Sybase does not recommend using `loginToSync`, because it blocks while waiting for network response. Instead, use `beginOnlineLogin`. This call sends the login message to Unwired Server and returns immediately. The application is then responsible for detecting login success or failure using the callback handler.

Solution: Report that Unwired server is down, or wait for it to come back online. Alternatively, redesign using `beginOnlineLogin`.

Explanation 2: The connection between the device and Unwired Server may be down. You can test to make sure the device is connected as required using the `SUPMessageClient` API:

```
while([SUPMessageClient status] != STATUS_START_CONNECTED)
    [NSThread sleepForTimeInterval:0.2];
[SampleApp_SampleAppDB beginOnlineLogin:@"supUser"
password:@"s3pUser"];
while([SampleApp_SampleAppDB getOnlineLoginStatus].status ==
SUPLoginPending)
{
    [NSThread sleepForTimeInterval:0.2];
}
// After this, the status will be either SUPLoginSuccess or
SUPLoginFailure
if([SampleApp_SampleAppDB getOnlineLoginStatus].status ==
SUPLoginSuccess)
    [SampleApp_SampleAppDB subscribe];
```

Solution: This reports the connection status. With a connection established, run this code on the device. After you run the code on the device, you can check from Sybase Control Center to see if the device is online.

Repeating a Query Produces No Results

Problem: On an iOS device, you cancel a query for a mobile workflow message, then repeat the same query. The second query does not produce results, and you see a message similar to `Error value is null or empty`.

Solution: Redesign the client application using the Mobile Workflow Forms Editor to prevent users from navigating back in the client application.

Windows Mobile and Windows

Provides troubleshooting information specifically for Windows Mobile and Windows device application clients.

Windows Mobile Device Synchronization Fails

Problem: Windows Mobile or Windows device synchronization fails.

Explanation: If the device synchronizes a replication-based synchronization (RBS) package with a deployed MBS package, the synchronization fails. The Unwired Server log will contain an error message indicating that the script version is not found.

```
Thread-141 [com.sybase.ml.sup.Logger] [-10013] Version
'default_flightmodelsample.1_0'
not found in the ml_script_version table. Cannot synchronize
```

If the device uses message-based synchronization (MBS) mode to subscribe to a deployed RBS package, the device application receives a 404 error indicating "Package not enabled for message-based synchronization."

Solution: Redeploy so the package types match.

Windows Mobile Device Cannot Sync

Problem: The Windows Mobile or Windows device cannot synchronize. Additional error messages and behavior differ for replication-based synchronization (RBS) and message-based synchronization (MBS).

Explanation: This may indicate the Unwired Server is down or unreachable.

- For RBS clients, if the device attempts to synchronize with the Unwired Server but the Unwired Server is down, the device receives a `SynchronizationException` indicating a `SOLE_COMMUNICATIONS_ERROR`.
- For MBS clients, the Sybase Messaging Client will keep retrying the request. The request will be handled when the Unwired Server restarts.

Solution: Report that Unwired Server is down, or wait for it to come back online.

Windows Mobile Device Cannot Establish Secure Connection

Problem: The Windows Mobile or Windows device client cannot set up a secure connection and reports a `SynchronizationException` message that indicates a communication error.

Explanation: One of the most common causes for not being able to set up a secure connection is an invalid certificate.

Solution: Make sure the certificate is valid, and the secure connection is set up correctly.

Cannot Test Database File on SD Card from Simulator

Problem: Cannot test a database file that is stored on an SD card from the Windows Mobile simulator.

Explanation: This is a known limitation of the Windows Mobile simulator. The simulator cannot test files on the SD card.

Solution: If you want to test the database file on the SD card, you must use a real device.

Failed to Create Component TextBox

Problem: Failed to create a component `TextBox` using drag and drop for a Windows Mobile client application. Error messages provide additional information.

```
Failed to create component 'TextBox'. The error message follows: 'System.IO.FileLoadException: Could not load file or assembly 'Microsoft.WindowCE.Forms, Version=3.5.0.0,Culture=neutral, PublicKeyToken =...'
```

Explanation: This indicates you must use a Windows command to complete the action.

Use the Windows command `gacutil /i Microsoft.WindowCE.Forms.dll` to add the assembly to the cache.



Validation Rules

Validation rules ensure that mobile business objects (MBOs) and the device applications and lightweight mobile workflow applications they interact with perform as expected in an enterprise environment. Error messages report problems and are useful for diagnosing the problem.

Mobile Business Object Validation Rules and Error Messages

Unwired WorkSpace (Eclipse) mobile business object (MBO) validation rules and error messages are divided into sections based on where in the MBO development cycle the validation rule applies.

Some errors and warnings are generated when you violate a validation rule, while others may occur during code generation, deployment, or when used within the device application, in which the error might not be obvious. Error information includes:

- Severity and message – there are two types of messages generated if a validation error occurs:
 -  Warning messages – A nonfatal error that requires attention, or a failure may occur at a later time.
 -  Error messages – the operation being performed fails. You must resolve the error.
 - Message - the actual error message, if any, that is generated if the validation rule is violated.
- Action – any additional details regarding the validation rule and how to resolve the error.

Note: You can identify some validation errors in Unwired WorkSpace by the warning or error icon that appears within the context of the error. For example:

- In the Properties view, if you add an unmapped parameter to an attribute, a warning icon and message appear. The same warning icon also appears in the header of the MBO in the Mobile Application Diagram.
- Mobile application project folders in WorkSpace Navigator display warning or error icons for invalid objects within the folder (MBOs, roles, personalization keys, and so on). You can isolate the error by expanding the folder until you reach the object with the error.

Other types of errors, for example deployment errors, generate messages.

In general, be aware of warning and error icons and messages, and resolve them early in the development cycle to ensure proper deployment, code generation, and device application function.

Mobile Business Object, Attribute, and Structure Object Error Messages and Validation Rules

These validation rules and error messages apply to mobile business objects (MBOs), attributes, and structure objects (complex object types), and are enforced during MBO development, and, in some cases, when you deploy the MBO to Unwired Server. Messages are also generated when validation rules are violated.

Table 8. MBO, attribute, and structure validation rules and error messages

Severity and message	Cause	Action
Error: Invalid mobile business object name: '<MBO Name>'. A valid identifier name must begin with an alphabetic character or an underscore, followed by alphanumeric characters or underscores.	The MBO name is invalid.	Enter a valid name.
Error: Invalid mobile business object name: '<MBO Name>', name cannot be a Java or C# language keyword.	The MBO name is a Java or C# keyword, such as binary, boolean, byte, char, date, datetime, time, decimal, double, float, object, int, integer, long, short, string, switch, and so on.	Do not use Java or C# keywords as MBO names.
Error: Invalid mobile business object name: '<MBO Name>', the length must be less than 64.	The length of the MBO name is longer than 64 characters.	Enter a name that is less than 64 characters.
Error: Invalid mobile business object name: '<MBO Name>', name cannot be a reserved attribute name.	The MBO name is a reserved attribute name: pending, replayPending, replayCounter, disableSubmit, pendingChange, lastModified, and deleted.	Enter a valid name.

Severity and message	Cause	Action
Error: Invalid mobile business object name: '<MBO Name>', name cannot be a reserved word.	The MBO name is a reserved MBO name: KeyGenerator, OperationReplay, ServerPersonalization, ClientPersonalization, SessionPersonalization, LogRecordImpl, and KeyGeneratorPK.	Enter a valid name.
Error: Invalid mobile business object name: '<MBO Name>', name cannot be a reserved internal type name.	The MBO name is an internal reserved type: "object", "binary", "boolean", "byte", "char", "date", "datetime", "decimal", "double", "float", "int", "integer", "long", "short", "string", "time"	Enter a valid name. For example, naming the MBO Date, DateTime, or Time results in an error when deploying the MBO.
Error: Duplicate mobile business object name: '<MBO Name>'. The name is case-insensitive.	An MBO with the same name already exists.	Enter a valid and unique name.
Error: Duplicate name: '<MBO Name>' in the object query.	The name of the MBO is the same as the name of an existing object query.	Enter a valid and unique name.
Error: The name: '<MBO Name>' duplicates with an operation name.	The name of the MBO is the same as the name of an existing operation. Validation occurs after creating the MBO after a table name change to an operation or attribute, then dragging-and-dropping the table to create the MBO.	Enter a valid and unique name.
Error: The name: '<MBO Name>' duplicates with an attribute name.	The name of the MBO is the same as the name of an existing attribute.	Enter a valid and unique name.
Error: Invalid structure name: '<Structure Name>'. A valid identifier name must begin with an alphabetic character or an underscore, followed by alphanumeric characters or underscores.	The structure type name is invalid.	Enter a valid name.

Severity and message	Cause	Action
<p>Error: Invalid structure name: '<Structure Name>'. A valid structure type name cannot be the same as the primitive type names.</p>	<p>The structure type name is a primitive datatype supported by Unwired WorkSpace. For example, int, boolean, and so on.</p>	<p>Enter a valid name.</p>
<p>Warning: Neither attribute nor operation is defined for the mobile business object: '<MBO Name>'.</p>	<p>An MBO exists without attributes or operations.</p>	<p>Define attributes, operations, or both for the MBO.</p>
<p>Warning: Attributes are not defined for the structure: '<Structure Name>'.</p>	<p>A structure exists without attributes.</p>	<p>Define attributes for the structure.</p>
<p>Error: Attributes are not defined for the mobile business object: '<MBO Name>'.</p>	<p>This error occurs in the MBO Creation wizard, in the "Attributes mapping" page if the wrong datasource operation definition is specified on the "Definition" page. In these cases, the data source operation cannot run successfully or there is no output record for the operation. No attributes are generated for the MBO.</p>	<p>In the MBO Creation wizard, do one of:</p> <ul style="list-style-type: none"> • Click Back to return to the "Parameters" page and verify that the correct default values are set. Correct any values as necessary then click Next to retry. • Click Back until you reach the "Definition" page and verify that the correct definition for the datasource operation is specified. Correct the problem and click Next, and specify default values for the parameters (arguments) on the "Parameters" page. Click Next to retry.
<p>Warning: Attributes are not bound to the back-end data source: '<MBO Name>'.</p>	<p>You can create an MBO without specifying a datasource, by selecting Bind data source later from the "Data source" page of the MBO Creation wizard. This warning message reminds you to bind the MBO to a data source at some point during the development process.</p>	<p>To bind an MBO to a data source after creating the MBO:</p> <ol style="list-style-type: none"> 1. Right-click the MBO in the Mobile Application Diagram, and select Show properties view. 2. From the Properties view, select the Attributes tab. 3. Select Data Source, and bind to a data source.

Severity and message	Cause	Action
Error: Invalid attribute name: '<Attr Name>'. A valid identifier name must begin with an alphabetic character or an underscore, followed by alphanumeric characters or underscores.	The attribute name is invalid.	Enter a valid name.
Error: Invalid attribute name: '<Attr Name>', name cannot be a Java or C# language keyword.	The attribute name is a Java or C# keyword, such as int, switch, and so on.	Do not use Java or C# keywords as attribute names.
Error: Invalid attribute name: '<Attr Name>', name cannot be a reserved attribute name.	The reserved keyword pending, replayPending, replayCounter, disableSubmit, pendingChange, lastModified, or deleted is used as an attribute name.	Enter a valid attribute name.
Error: Invalid attribute name: '<Attr Name>', the length must be less than 64.	The attribute name exceeds the 64 character maximum length.	Shorten the attribute name.
Error: Duplicate attribute name: '<Attr Name>'. The name is case-insensitive.	An attribute with the same name already exists.	Enter a valid name.
Error: Duplicate name: '<Attr Name>' in the mobile business object.	The attribute name is the same as the MBO name to which it belongs.	Enter a valid name.
Error: Duplicate name: '<Attr Name>' in the object query.	The attribute name is the same as an object query name for a given MBO.	Enter a valid name.
Error: The name: '<Attr Name>' duplicates with an operation name.	The attribute name is the same as an operation name for a given MBO.	Enter a valid name.
Error: Invalid attribute name: '<Attr Name>', name cannot be a reserved word.	A reserved keyword is used as an attribute name: deleted, lastAccessed, lastModified, isSubscribed, pending or hashCode.	Enter a valid name.

Validation Rules

Severity and message	Cause	Action
Warning: Attribute '<Attr Name>' has no mapped data source field and no propagated load parameter.	An attribute is not mapped to a column or to a load parameter. This warning is generated if neither is mapped.	In the Properties view, set either the column mapping or load parameter mapping. To map a load parameter, set Propagate to on the Load parameters tab.
Warning: Data type of the column cannot map to the attribute: '<Attr Name>'.	Datatype mismatch.	Change the attribute datatype or column datatype (if it is not read-only) so they match.
Warning: Attribute '<Attr Name>' should be nullable when it's mapped to a nullable column.	The attribute is not nullable, but the column to which it maps is. The warning indicates that a failure might occur when loading or synchronizing data based on this attribute.	Change the nullability of the attribute or the column to which it maps so they match.
Warning: Attribute data length is shorter than the mapped column length, value truncation might occur: '<Attr Name>'.	The datatype length of the attribute is shorter than the back-end column length. The value might be truncated when loading data from the back-end datasource to the consolidated database during refresh.	Change the attribute's length to match that of the column to which it maps.
Warning: Attribute binary data length is shorter than the mapped column length, NULL will be returned: '<Attr Name>'.	The datatype length of the binary type attribute is shorter than the backend column length, so the wrong NULL value could be loaded from the backend datasource to the consolidated database during refresh.	Change the attribute's length to match that of the column to which it maps.

Severity and message	Cause	Action
Error: Attribute '<Attr Name>' of type {0}, with length greater than {2} bytes should not be used as a primary key.	<p>The attribute is set as a primary key, but its length is longer than the primary key's length allowed by the server:</p> <ul style="list-style-type: none"> • {0} may be STRING or BINARY • {2} 536870912(2G/4) for STRING • 0x7fffffff (Integer.MAX_VALUE, 2G-1) for BINARY 	Change the primary key attribute's length to a valid value as indicated by the error message.
Error: Attribute '<Attr Name>' of type {0}, does not allow length greater than {2}.	<p>The attribute is longer than the allowed maximum length:</p> <ul style="list-style-type: none"> • {0} may be STRING or BINARY • {2} 536870912(2G/4) for STRING • 0x7fffffff (Integer.MAX_VALUE, 2G-1) for BINARY 	Change the attribute length to a valid value as indicated by the error message.
Error: Column '<Col Name>' of type {0}, does not allow length greater than {2}.	<p>The column is longer than the allowed maximum length:</p> <ul style="list-style-type: none"> • {0} may be STRING or BINARY • {2} 536870912(2G/4) for STRING • 0x7fffffff (Integer.MAX_VALUE, 2G-1) for BINARY 	Change the attribute length to a valid value as indicated by the error message.

Severity and message	Cause	Action
Warning: The relationship between source mobile business object '<Source MBO Name>' and target mobile business object '<Target MBO Name>' may not work, because the load parameter of target mobile business object is mapped to both attribute of source mobile business object and synchronization parameter of target mobile business object.	If a load parameter is bound to both a foreign attributes and a synchronization parameter, then the relationship between the source and target MBOs may not work.	Select the correct mapping for load parameters according to a valid scenario.
Error: All load parameter(s) of target mobile business object '<MBO Name>' must be filled either from attribute(s) of source mobile business object(s) or from synchronization parameter(s) of target mobile business object, a combination is not allowed.	If an MBO has multiple load parameters, then they can all come from sync parameters, or all from "foreign attributes" (for example, attributes of a parent MBO), but a mixture is not supported. Otherwise, deployment will fail.	Ensure that load parameters of MBOs come from sync parameters, or that all come from "foreign attributes".
Error: Mobile business object "<MBO Name>" with online cache policy must have at least one load parameter with "Propagate To" attribute set.	Online MBO does not have a load parameter with "Propagate To" set.	Assign a "Propagate To" attribute for at least one load parameter.

Parameter and Argument Validation Rules and Error Messages

These validation rules and error messages apply to mobile business object parameters and arguments, and are enforced during MBO development, and, in some cases, when you deploy the MBO to Unwired Server. Messages are generated when validation rules are violated.

Table 9. Parameter and argument validation rules and error messages

Severity and message	Cause	Action
Error: Invalid parameter name: '<Parameter Name>'. A valid identifier name must begin with an alphabetic character or an underscore, followed by alphanumeric characters or underscores.	The parameter name is invalid.	Enter a valid name.

Severity and message	Cause	Action
Error: Invalid parameter name: '<Para Name>', name cannot be a Java or C# language.	The parameter name is a Java or C# keyword, such as binary, boolean, byte, char, date, datetime, time, decimal, double, float, object, int, integer, long, short, string, switch, and so on.	Do not use Java or C# keywords as parameter names.
Error: Invalid parameter name: '<Para Name>', the length must be less than 100.	The length of the parameter name is longer than 100 characters.	Shorten the parameter name to less than 100 characters.
Error: Duplicate parameter name: '<Para Name>'.	The name of the parameter (case insensitive) is the same as an existing parameter.	Enter a valid and unique name.
Warning: No parameter mapped to argument: '<Arg Name>'	A complex type has at least one parameter within it that is not mapped to an argument.	Map all parameters within a complex type to arguments.
Error: The argument '<Arg Name>' in operation '<Oper Name>' is not nullable and does not have a parameter mapped, thus it must have a default value.	A default value is not specified for the operation's parameter, which may result in a runtime operation failure.	Specify a default value for the argument.
Error: Attribute '<Attr Name>' cannot be mapped to more than one parameter.	An attribute has more than one Propagate to mapping.	Remove additional mappings.
Error: The data type of the parameter '<Para Name>' can not map to the data type of the argument.	The datatype of the parameter and the datatype of the argument do not match.	Change the datatype of the parameter to match that of the argument. Or change the type of the argument if it is not read-only. <hr/> Note: Make sure the argument type matches that of the data source for JDBC data sources. <hr/>
Warning: Parameter '<Para Name>' should not be nullable when it is mapped to a non-nullable argument.	The nullability of the parameter does not match that of the argument. For a non-nullable argument, the mapping parameter should also be non-nullable.	Change the parameter to non-nullable.

Validation Rules

Severity and message	Cause	Action
Warning: Parameter data length is larger than the mapped argument length, value truncation might occur: '<Para Name>'. 	The datatype length of the parameter does not match that of the argument.	Change the datatype length of the parameter or that of the argument. Note: Unwired WorkSpace allows the parameter datatype length to be shorter than the length of the argument.
Warning: The default value of parameter '<Para Name>' should not be set to '<NULL>' when its 'Fill from attribute' '<Attr Name>' is a primary key.	A null value is assigned to the parameter, and the parameter's Fill from attribute is a primary key.	Assign a non-null value to the parameter.
Warning: The default value of parameter '<Para Name>' should not be set to '<NULL>' when it is not used in target mobile business object and its 'Propagate to' attribute '<Attr Name>' is a primary key and not mapped.	This warning occurs under these conditions: <ol style="list-style-type: none"> 1. The load parameter is not used in the target MBO in a relationship. 2. The load parameter has a null default value. 3. The Propagate to attribute of the load parameter is a primary key and is not mapped to a data source column. 	Assign a non-null default value to the load parameter.
Warning: Non-updatable parameter should have 'Fill From Attribute' defined: '<Para Name>'. 	A nonupdatable parameter does not have a Fill from attribute setting.	Set the Fill from attribute for the parameter.
Error: The default value of a non-updatable parameter that has no 'Fill from Attribute' or personalization key should be set for 'CREATE' type operation: '<Oper Name>'. 	A create operation has a parameter that does not have a Fill from attribute or a personalization key set.	Specify a default value for the parameter.
Error: The default value of a non-updatable parameter that has no personalization key should be set for 'OTHER' type operation: '<Oper Name>'. 	An other operation parameter does not have a personalization key set.	Specify a default value for the parameter.

Severity and message	Cause	Action
Error: Data type of the bound personalization key cannot map to the parameter: '<Para Name>'.	Datatype mismatch.	Change the datatype of the parameter or the datatype of the personalization key so they match.
Warning: Parameter data length is shorter than the bound personalization key's data length, value truncation might occur: '<Para Name>'.	The length of the datatypes do not match.	Change the length of the parameter or that of the personalization key so they match.
Warning: Parameter '<Para Name>' should not be bound to personalization key '<PK Name>' whose default value is set to '' when its 'Propagate to' attribute '<Attr Name>' is a primary key.	The parameter is bound to a misconfigured personalization key.	Redefine the personalization key according to the message, or remove the mapping.
Error: The nullability of the parameter '<Para Name>' should be the same as the personalization key: '<PK Name>'.	Parameter and personalization key nullability mismatch.	Change the nullability of the parameter or that of the personalization key.
Error: Data type of the parameter '<Para Name>' must be the same as that of its selected synchronization parameter.	Datatype mismatch.	Change the datatype of the parameter or that of the synchronization parameter so they match, or map the parameter to a different synchronization parameter.
Warning: Parameter data length is shorter than the 'Synchronization Parameter' length, value truncation might occur: '<Para Name>'.	The datatype lengths do not match. At runtime, when the synchronization parameter passes its value to the load parameter, the value may be truncated.	Change the length of either parameter so they match.
Error: Data type of the attribute cannot map to the parameter: '<Para Name>'.	Datatype mismatch.	Change the datatype of the parameter or that of the attribute so they match, or map to another appropriate attribute.
Warning: Parameter data length is larger than that of the attribute it propagates to, value truncation might occur: '<Para Name>'.	The datatype lengths do not match. At runtime, when the parameter passes its value to the propagated attribute, the value may be truncated.	Change the datatype length so they match.

Validation Rules

Severity and message	Cause	Action
Warning: The maximum length of the operation parameter is different from the fill from attribute's: '<Para Name>'.	The datatype lengths do not match. At runtime, when the attribute passes its value to the parameter, the value may be truncated.	Change the datatype length so they match.
Error: Binary parameter default value cannot exceed 16384 byte: '<Para Name>'.	The default value is too long.	Shorten the default value.
Warning: Association target <MBO Name>'s parameter '<Para Name>' must have personalization key or default value defined when it is not a target parameter of an association.	There is no source from which the parameter value can be set.	Set a default value or specify a personalization key for the parameter.
Error: Multiple parameters cannot map to the same argument '<Arg Name>'.	Two or more parameters map to the same argument.	Change the mapping to allow only one parameter to map to the argument.
Warning: Default value, personalization key, or synchronization parameter is not defined for the parameter: '<Para Name>'.	There is no source from which the load parameter value can be set.	Specify the value of the load parameter from a default value, a personalization key, or a synchronization parameter.
Error: Synchronization parameter '<Sync Para Name>' cannot be mapped to more than one parameter.	Two or more parameters map to the same synchronization parameter.	Change the mapping so only one parameter maps to the synchronization parameter.
Warning: Non-nullable parameter '<Para Name>' cannot map to a nullable synchronization parameter '<Sync Para Name>'.	There is a nullability mismatch between the parameter and the synchronization parameter to which it maps.	Change the nullability of either so they match.
Error: Data type of the selected old value argument cannot map to the parameter '<Para Name>' of operation '<Oper Name>'.	Datatype mismatch.	Change the datatype of the Old value argument or that of the parameter so they match.
Warning: The old value argument '<Arg Name>' will be ignored when the parameter '<Para Name>' has no mapped argument.	The parameter does not have an argument mapping, and is mapped to an Old value argument .	Specify the argument mapping for the parameter.

Severity and message	Cause	Action
Warning: No argument is mapped to parameter '<Para Name>'.	The parameter of the operation is not mapped to an argument.	Map the parameter to an argument.
Warning: No argument is mapped to parameter '<Para Name>'.	The load parameter is not mapped to an argument.	Map the load parameter to an argument.
Error: Argument '<Arg Name>' of type <Type>, does not allow length greater than <Arg Length>.	The argument length exceeds the maximum length allowed for its datatype.	Shorten the argument length to a valid length for the given type according to the message.
Error: Invalid default value '<Default Value>' for the '<Type>' data type: '<Arg Name>'.	The argument default value is not valid for its datatype.	Change the default value to conform to the argument's datatype.
Warning: Default value exceeds maximum length: '<Arg Length>'.	The argument default value length exceeds the argument's length.	Change the default value to conform to the argument's length.
Error: Invalid structure default value for the '<Type>' data type of argument '<Arg Name>': '<Msg>'.	The structure argument default value contains errors.	Correct the default value using the Edit values dialog according to the error message.
Error: Personalization key can't be directly assigned to load parameter: '<Para Name>' of mobile business object which has a scheduled cache policy.	The load parameter is directly assigned a personalization key while the MBO is in a cache group that uses a scheduled cache policy.	Move the MBO to a cache group that does not use a scheduled cache policy, or remove the personalization key from the load parameter.
Warning: The nullability of the bound personalization key '<personalization key name>' is not consistent with parameter '<parameter name>'.	The nullability between personalization key and parameter is inconsistent.	Specify the same nullability support for the personalization key and parameter.

Operation Validation Rules and Error Messages

Validation rules are enforced during MBO development, and, in some cases, when you deploy the MBO to Unwired Server. Messages are generated when validation rules are violated.

Table 10. MBO operation validation rules and error messages

Severity and message	Cause	Action
Error: Invalid operation name: '<Oper Name>'. A valid identifier name must begin with an alphabetic character or an underscore, followed by alphanumeric characters or underscores.	The operation name is invalid.	Enter a valid name.
Error: Invalid operation: '<Oper Name>', name cannot be a Java or C# language keyword.	The operation name is a Java or C# keyword, such as binary, boolean, byte, char, date, datetime, time, decimal, double, float, object, int, integer, long, short, string, switch, and so on.	Do not use Java or C# keywords as operation names.
Error: Invalid operation name: '<Oper Name>', the length must be less than 64.	The length of the operation name is longer than 64 characters.	Enter a valid name shorter than 64 characters.
Error: Duplicate operation name: '{0}'. The name is case-insensitive.	Duplicate operation name.	Enter a valid and unique name.
Error: Duplicate name: '<Oper Name>' in the mobile business object.	The operation name is the same as the MBO name to which it belongs.	Enter a valid name.
Error: Duplicate name: '<Oper Name>' in the object query.	The operation name is the same as an object query for this MBO.	Enter a valid name.
Error: The name: '<Oper Name>' duplicates with an attribute name.	The operation name is the same as an attribute name for the MBO.	Enter a valid name.
Warning: Operation name '<Oper Name>' starts with 'findby', which may cause name conflict when generating client code.	The operation name cannot start with "findby."	Enter a valid name.

Severity and message	Cause	Action
Warning: Operation is not bound to the back-end data source: '<Oper Name>'.	The operation is not bound to a data source.	Bind a data source to the operation.
Error: If a mobile business object has no attributes, the operation '<Oper Name>'s type must be set to 'OTHER'.	An operation type other than "other" has been defined for the MBO, and no attributes are defined.	Only "other" operations support no attributes. Change the operation type to "other", or add attributes to the MBO.
Error: Parameter '<Para Name>' cannot have 'Fill from attribute' as the type of the operation '<Oper Name>' is 'OTHER'.	An "other" operation parameter has a Fill from attribute mapping set.	Since "other" operations are static from a code generation perspective, remove all Fill from attribute mappings from "other" operation parameters.
Error: The response representation of the "UPDATE" operation '<Operation Name>' with "Apply results to the cache" cache update policy does not have an XSLT defined.	For restful Web service data sources, If an MBO update operation does not have an XSLT associated with it and the "Apply results to the cache" cache update policy is set, after deployment to the server, NullPointerException will be thrown during the execution.	Add an XSLT for the response representation.
Error: The "UPDATE" operation "<Operation Name>" with "Apply results to the cache" cache update policy does not have XSLT.	For SOAP or WSDL based data sources, If an MBO update operation does not have an XSLT associated with it and the "Apply results to the cache" cache update policy is set, after deployment to the server, NullPointerException will be thrown during execution.	Add an XSLT for update operation.

Relationship Validation Rules and Error Messages

Validation rules are enforced when defining a relationship between two MBOs and, in some cases, when you deploy the MBO to Unwired Server. Messages are generated when validation rules are violated.

Table 11. Relationship validation rules and error messages

Severity and message	Cause	Action
Error: Duplicate relationship name: '<Relationship Name>' case insensitive.	A duplicate relationship name (source attribute name) is defined.	Change the autogenerated name of the source attribute in the Relationship Creation wizard, Edit dialog, or Properties view.
Error: No mapping information specified for the relationship: '<Relationship Name>'.	The relationship does not have a mapped attribute or mapped parameter.	Map the attribute or parameter in the Relationship Creation wizard, Edit dialog, or Properties view.
Error: Relationship name '<Relationship name>' is a duplicate of the existing attribute name '<Attribute name>'.	The relationship's name is a duplicate of the source MBO attribute's name.	Change the attribute name in the source MBO, or change the source attribute's relationship name.
Error: Relationship target attribute name '<Attribute name>' is a duplicate of the existing attribute name '<Attribute name>'.	The relationship's target attribute name is a duplicate of an attribute name in the target MBO.	Change the target MBO attribute name, or change the target attribute's name.
Error: Relationship target attribute name '<Attribute name>' is a duplicate of the existing source attribute name '<Attribute name>'.	The relationship's target attribute name is a duplicate of the source attribute's name in another relationship.	Change this relationship's target attribute name, or change the duplicated source attribute name.
Error: Attributes must be defined for '<MBO Name>' to create a relationship.	No attributes are defined in the source MBO or the target MBO.	Create attributes in the source or target MBO.

Severity and message	Cause	Action
<p>Error: Primary key attributes must be defined for '<MBO Name>' in the relationship '<Relationship name>'.</p>	<p>Primary key attributes are not defined in the source MBO or the target MBO.</p>	<p>Set the primary key attribute in the source MBO in one-to-many bidirectional, unidirectional, and one-to-one bi-directional relationships.</p> <p>Set the primary key attribute in the target object (MBO) in many-to-one bidirectional, unidirectional, and one-to-one unidirectional relationships. You can either:</p> <ul style="list-style-type: none"> • Set the related attribute as primary key on source or target MBOs according to the relationship's type. • automatically set or unset the mapped attribute as a primary key preference setting (from the Preferences menu, select Sybase, Inc > Mobile Development > Miscellaneous > Relationship). • Map the attribute or parameter between the source and target MBO.
<p>Error: The primary key attribute '<Attribute name>' must be mapped in the relationship '<Relationship name>'.</p>	<p>There is no primary key mapping for the source or target MBO attribute.</p>	<p>Unset the attribute as a non-primary key, or map the attribute in the relationship. For one-to-many bidirectional, unidirectional and one-to-one bidirectional relationships, map all primary key attributes of the source MBO. For many-to-one bidirectional, unidirectional and one-to-one unidirectional relationships, map all primary key attributes of the target MBO.</p>

Validation Rules

Severity and message	Cause	Action
Error: The non-primary key attribute '<Attr Name>' should not be mapped in the relationship '<Relationship Name>'.	At least one non-primary key attribute is mapped in the relationship.	Set the mapped attribute as a primary key, or unmap the attribute in the relationship. For one-to-many bidirectional, unidirectional and one-to-one bi-directional relationships, unmap all non-primary key attributes of the source MBO. For many-to-one bidirectional, unidirectional and one-to-one unidirectional relationships, unmap all non-primary key attributes of the target MBO.
Error: The data type of attribute 'Attribute name' does not match 'Attribute name' in the relationship.	The datatype of the mapped attribute/parameter does not match.	Map the attribute/parameter to a target with the same datatype.
Error: Cannot create relationship between different type of mobile business object.	A relationship between different types of MBOs exists.	Create relationships only between the same types of MBOs.
Error: Cannot create relationship between structures.	A relationship between structures exists.	Relationships between structures is not currently supported.
Error: The target mobile business object '<MBO Name>' can not be used in the composite relationship 'relationship name' since it already was used in a composite relationship as a target.	One MBO is used in multiple composite relationships as a target MBO.	In composite relationships, define only one MBO as a target MBO.
Error: The target mobile business object '<MBO Name>' should have a DELETE type operation in the composite relationship '<Relationship name>'.	There is no delete operation in the target MBO of a composite relationship.	Add a delete operation to the target MBO.
Error: The parameter '<Parameter name>' should set the propagate-to attribute in the relationship between '<MBO Name>' and '<MBO Name>'.	The mapped load parameter does not have a propagate-to attribute in the relationship.	Unmap the load parameter, or set the propagate-to attribute for the load parameter.

Severity and message	Cause	Action
<p>Error: As the associated propagate-to attribute is a primary key, the parameter '<Parameter name>' should be mapped to an attribute or a parameter in the relationship with '<MBO Name>'.</p>	<p>The load parameter, which has a propagate-to attribute primary key, is not mapped in the relationship.</p>	<p>Unset the propagate-to attribute as non-primary key. Or,</p> <p>Map the load parameter in the relationship for both the target and source MBO: for one-to-many bidirectional, unidirectional and one-to-one bidirectional relationship, map all load parameters where the propagate-to attribute are primary key attributes of the source MBO. For many-to-one bidirectional, unidirectional and one-to-one unidirectional relationship, map all load parameters where the propagate-to attribute are primary key attributes of the target MBO.</p>
<p>Error: The parameter '<Parameter name>' was mapped in the relationship '<relationship name>'. Its propagate-to attribute '<Attribute name>' must be a primary key.</p>	<p>The load parameter's propagate-to attribute is mapped in the relationship, but is not a primary key.</p>	<p>Set the propagate-to attribute as a primary key, or Unmap the load parameter in the relationship:</p> <ul style="list-style-type: none"> • Unmap all propagate-to attribute load parameters that are non-primary key attributes in the source MBO if used in one-to-many bidirectional, unidirectional, and one-to-one bidirectional relationships. • Unmap all propagate-to attribute load parameters that are non-primary key attributes in the target object (MBO) if used in many-to-one bidirectional, unidirectional, and one-to-one unidirectional relationships.

Severity and message	Cause	Action
<p>Error: The primary key attribute '<Attribute name>' was not used in the relationship '<Relationship name>'.</p>	<p>The primary key attribute is not mapped in the relationship, but is mapped in another relationship.</p>	<p>Map the attribute in the relationship:</p> <ul style="list-style-type: none"> • Map all primary key attributes of the source MBO in one-to-many bidirectional, unidirectional, and one-to-one bidirectional relationships. • Map all primary key attributes of the target MBO in many-to-one bidirectional, unidirectional, and one-to-one unidirectional relationships.
<p>Error: The propagate-to attribute of the parameter '<Parameter name>' is a primary key and must be used in the relationship '<Relationship name>'.</p>	<p>The propagate-to attribute load parameter, which is a primary key, is not mapped in the current relationship, but is mapped in another relationship.</p>	<p>Map the load parameter in the relationship:</p> <ul style="list-style-type: none"> • Map all propagate-to attribute load parameters that are primary key attributes of the source MBO in one-to-many bidirectional, unidirectional, and one-to-one bidirectional relationships. • Map all propagate-to attribute load parameters that are primary key attributes of the target MBO in many-to-one bidirectional, unidirectional, and one-to-one unidirectional relationships.
<p>Error: Relationship from 'MBO Name' to '<MBO Name>' is invalid, as it causes a circular relationship.</p>	<p>The relationship is circular.</p>	<p>Remove the relationship. Circular relationships are not allowed.</p>
<p>Error: Relationship from '<MBO Name>' to '<MBO Name>' references two mobile business objects in different synchronization groups. Mobile business objects with relationship should be grouped in the same synchronization group.</p>	<p>The two MBOs in the relationship do not belong to the same synchronization group.</p>	<p>Move the source and target MBOs into the same synchronization group.</p>

Severity and message	Cause	Action
Error: Composite relationship from '<MBO Name>' to '<MBO Name>' references two mobile business objects in different synchronization groups. Mobile business objects with composite relationship should be grouped in the same synchronization group.	The two MBOs in a composite relationship are not in the same synchronization group.	Move the source and target MBOs into a same synchronization group.
Error: Relationship '<relationship name>' references two mobile business objects in different cache groups. Mobile business objects with relationship should be put in the same cache group.	The two MBOs in the relationship do not belong to the same cache group.	Move the source and target MBOs into a same cache group.
Error: Composite relationship "<relationship_name>" references two mobile business objects in different cache groups. Mobile business objects with composite relationship must be put in the same cache group.	The two MBOs in a composite relationship are not in the same cache group.	Move the source and target object (MBO) into the same cache group.
Error: The secondary mobile business object '<MBO Name>' can not be used in the relationship '<Relationship name>' as source object.	In a relationship that involves multiple MBOs created from the same EIS operation, the secondary (subsequent) MBO is the relationship's source object and the target MBO is its primary MBO, and the load parameter of the primary MBO is mapped.	Do not map the load parameter of the source MBO, or swap the MBOs so that the secondary MBO is the primary MBO.
Error: The secondary mobile business object '<MBO Name>' can not be used in the relationship '<Relationship name>' as target object.	In a relationship that involves multiple MBOs created from the same EIS operation, the secondary MBO is the target object and its load parameter is mapped.	Do not map the load parameter of the target object (MBO) unless the source MBO is also the primary MBO, or swap the MBOs so that the secondary MBO is the primary MBO.

Severity and message	Cause	Action
Error: Multiple relationships between "<MBO1_name>" and "<MBO2_name>" caused circular dependencies, must set one side of the foreign key attributes to nullable.	The multiple relationships between MBOs form circular dependencies with foreign keys non-nullable in both MBO. Unwired Server cannot load MBOs for this case.	Make the foreign key attributes nullable in either source or target side.
Error: Relationship "<Relationship name>" involves mobile business objects with online cache policy must be a composite relationship.	The relationship in online cache group is not a composite relationship.	Make the relationship a composite relationship.
Error: Relationship has two mobile business objects with independent load definitions in the online cache group, and this is not allowed.	The relationship is established between non-commingled MBOs in online cache groups.	Remove the unsupported relationship.
Error: Relationship "<Relationship Name>" references mobile business objects across "<Cache group type1>" and "<Cache group type2>" cache groups.	Mobile business objects with relationship should be put in the same cache group. Relationship established on MBOs across online and non-online cache groups.	Make the source and target MBOs cache groups are consistent—either all in an online cache group, or all in a non-online cache group.

Cache Validation Rules and Error Messages

These validation rules apply to cache groups and cache update policies and are enforced during MBO development and, in some cases, when you deploy the MBO to Unwired Server. Messages are generated when validation rules are violated.

Table 12. Cache group and cache update policy validation rules and error messages

Severity and message	Cause	Action
Error: Invalid interval for cache '<CacheGrp Name>': the value of interval must be between 1 and 2147483647 seconds.	The cache group's cache interval is out of range.	Specify a valid cache interval.
Error: Cache group name cannot be empty.	The cache group name is empty.	Enter a valid name.

Severity and message	Cause	Action
Error: Invalid cache name: '<CacheGrp Name>'. A valid identifier name must begin with an alphabetic character or an underscore, followed by alphanumeric characters or underscores.	An invalid cache name is specified.	Enter a valid name.
Error: Invalid cache name: '<CacheGrp Name>', name cannot be a Java or C# language.	The cache group name is a Java or C# keyword, such as binary, boolean, byte, char, date, datetime, time, decimal, double, float, object, int, integer, long, short, string, switch, and so on.	Do not use Java or C# keywords as cache group names.
Error: Invalid cache name: '<CacheGrp Name>', the length must be less than 64.	The length of the cache group name is longer than 64 characters.	Enter a valid name that is shorter than 64 characters.
Error: Duplicate cache group name: '<CacheGrp Name>'.	Another cache group uses the same name.	Enter a unique and valid name.
Warning: Relationship from 'MBO1' to 'MBO2' references to different cache groups. Mobile business objects with relationship should be grouped in the same cache group.	MBOs in a relationship are in different cache groups. This is allowed, but be aware that MBOs in different cache groups may use different cache refresh policies.	Move the MBOs within a relationship to the same cache group if they need to share the same data refresh schedule.
Warning: 'CREATE' operation '<Oper Name>' has 'Apply results to the cache' option set without returning primary key fields.	A create operation uses the "Apply results to the cache" cache update policy, but the MBO to which it belongs does not have a primary key.	Set a primary key for the MBO in the Properties view from the Attributes > Attributes Mapping tab.

Severity and message	Cause	Action
Warning: The output record of the 'CREATE' operation '<Oper Name>' with 'Apply results to the cache' option set does not contain the primary key columns.	The output of the create operation that uses the Apply results to the cache cache update policy does not contain the primary key columns.	Change the definition of the Create operation so the output covers the MBO primary key columns. Note: This rule applies only to non-JDBC data sources. For JDBC data sources, since Unwired WorkSpace cannot identify the output metadata of an operation without executing it, you must verify that you have correctly defined the operation.

Synchronization Validation Rules and Error Messages

These validation rules apply to synchronization parameters and are enforced during MBO development, and, in some cases, when you deploy the MBO to Unwired Server or synchronize a device application. Messages are generated when validation rules are violated.

Table 13. Synchronization validation rules and error messages

Severity and message	Cause	Action
Error: Data type of the bound personalization key cannot map to the parameter: '<Para Name>'.	The datatype of the mapped personalization key differs from the datatype of the synchronization parameter, which may result in a synchronization failure.	Change the datatype of the personalization key to match the datatype of the synchronization parameter to which it maps.
Warning: Parameter data length is shorter than the bound personalization key's data length, value truncation might occur: '<Para Name>'.	Datatype length mismatch.	Change the datatype length of the parameter or that of the personalization key so they match.
Error: The nullability of the parameter '<Para Name>' should be same with the personalization key: '<PK Name>'.	The nullability of the parameter and the personalization key to which it maps do not match.	Change the nullability of the synchronization parameter or that of the personalization key so they match.
Error: Cannot specify 'Query Limiting' for synchronization parameter '<Para Name>' with list type personalization key.	The Query limiting option is selected for a parameter that has a list type personalization mapping.	Unselect Query limiting , which does not support mapping of list type personalization keys.

Severity and message	Cause	Action
Error: Invalid default value '<Default value>' for the '{1}' data type: '<Para Name>'.	{1} is an invalid datatype name.	Specify a valid default value.
Warning: Default value exceeds maximum length: '<Para Name>'.	The default value is longer than 100 characters.	Enter a default value that is less than 100 characters.
Error: Binary parameter default value cannot exceed 16384 byte: '<Para Name>'.	The default value length is more than 16384 bytes.	Shorten the default value.
Error: The data types of the parameter '<Para Name>' and the 'Mapped to' attribute '<Attr Name>' should match.	Datatype mismatch.	Change the datatype of the synchronization parameter, or map it to a different attribute, or change the datatype of the attribute to which it maps.
Warning: The nullability of the parameter '<Para Name>' should be the same as the attribute it is mapped to.	The nullability of the parameter and the attribute to which it maps do not match.	Change the nullability of the synchronization parameter or that of the attribute so they match.
Warning: The data length of the parameter '<Para Name>' is different from the data length of the its 'Mapped to' attribute '<Attr Name>'.	Datatype length mismatch.	Change the datatype length of the synchronization parameter or that of the attribute so they match.
Warning: No 'Mapped to' attribute specified for synchronization parameter: '<Para Name>'.	There is no Map to attribute defined for the parameter, which may lead to a synchronization failure.	Specify a Map to attribute for the synchronization parameter.
Error: Mobile business object '<MBO Name>' has more than one synchronization parameters that are mapped to multiple personalization keys.	More than one synchronization parameter is mapped to multiple list type personalization keys.	Remove any extra list type personalization key mappings. Only one synchronization parameter can be mapped to a list type personalization key.
Error: Synchronization parameter '<Para Name>' of type {0}, does not allow length greater than {2}.	The synchronization parameter length is too long: {0} may be STRING or BINARY {2} 8191 for STRING 32767 for BINARY	Shorten the datatype length.

Severity and message	Cause	Action
Error: Invalid parameter name: '<Para Name>'. A valid identifier name must begin with an alphabetic character or an underscore, followed by alphanumeric characters or underscores.	An invalid parameter name is specified.	Enter a valid name.
Error: Invalid parameter name: '<Para Name>', name cannot be a Java or C# language.	The parameter name is a Java or C# keyword, such as binary, boolean, byte, char, date, datetime, time, decimal, double, float, object, int, integer, long, short, string, switch, and so on.	Do not use Java or C# keywords as parameter names.
Error: Invalid parameter name: '<Para Name>', the length must be less than 100.	The length of the parameter name is longer than 100 characters.	Enter a name shorter than 100 characters.
Error: Duplicate parameter name: '<Para Name>'.	The name of the parameter is the same as an existing synchronization parameter.	Enter a valid name.
Error: Object query cannot be empty.	A Customized download data query is empty.	Select Customized download data from the Synchronization tab in the Properties view, and define a valid object query. Note: Unwired WorkSpace validates only that the query returns a result, not the syntax. You must verify that the query syntax is correct.

Object Query Validation Rules and Error Messages

Validation rules are enforced during object query development, and, in some cases, when you deploy the MBO to Unwired Server. Messages are generated when validation rules are violated.

Table 14. Object query validation rules and error messages

Severity and message	Cause	Action
Error: Object query cannot have an empty name.	An empty string is specified as the object query name.	Enter a valid name.

Severity and message	Cause	Action
Error: Invalid object query name: '<Query Name>'. A valid identifier name must begin with an alphabetic character or an underscore, followed by alphanumeric characters or underscores.	An invalid object query name exists.	Enter a valid name.
Error: Invalid object query name: '<Query Name>', name cannot be a Java or C# language.	The object query name is a Java or C# keyword, such as binary, boolean, byte, char, date, datetime, time, decimal, double, float, object, int, integer, long, short, string, switch, and so on.	Do not use Java or C# keywords as object query names.
Error: Invalid object query name: '<Query Name>', the length must be less than 64.	The length of the object query name is longer than 64 characters.	Shorten the object query name.
Error: Duplicate object query name: '<Query Name>'.	The object query name is a duplicate.	Enter a valid and unique name.
Error: Duplicate name: '<Query Name>' in the mobile business object.	The object query name is the same as the MBO to which it belongs.	Enter a valid and unique name.
Error: The name: '<Query Name>' duplicates with an operation name.	The object query name is the same as an operation in an MBO to which they both belong.	Enter a valid and unique name.
Error: The name: '<Query Name>' duplicates with an attribute name.	The object query name is the same as an attribute in an MBO to which they both belong.	Enter a valid and unique name.
Error: Query name '<Query Name>' is a reserved word.	The object query name is a reserved word: pull, downloadData, find, or load.	Enter a valid name.

Severity and message	Cause	Action
Error: Query definition cannot be empty.	The object query definition is empty.	Define a valid object query. Note: Unwired WorkSpace validates only whether the object query is empty, but does not validate its syntax.
Warning: No attribute is mapped to parameter '<Para Name>'.	The object query parameter has no attribute mapping.	Specify an attribute mapping for the parameter.

Personalization Key Validation Rules and Error Messages

Validation rules are enforced when defining personalization keys, assigning them to attributes or parameters, and, in some cases, when you deploy the MBO to which the personalization key is assigned to Unwired Server. Messages are generated when validation rules are violated.

Table 15. Personalization key validation rules and error messages

Severity and message	Cause	Action
Error: Invalid personalization key name: '<PK Name>'. A valid identifier name must begin with an alphabetic character or an underscore, followed by alphanumeric characters or underscores.	The personalization key name is invalid.	Enter a valid name.
Error: Invalid personalization key name: '<PK Name>', name cannot be a Java or C# language keyword.	The personalization key name is the same as a Java or C# keyword, such as binary, boolean, byte, char, date, date-time, time, decimal, double, float, object, int, integer, long, short, string, switch, and so on.	Do not use Java or C# keywords as personalization key names.
Error: Invalid personalization key name: '<PK Name>', the length must be less than 64.	The length of personalization key name exceeds the allowed maximum.	Enter a valid name that is less than 64 characters.
Error: Duplicate personalization key name: '<PK Name>'. The name is case-insensitive.	A personalization key with the same name already exists.	Enter a valid and unique name.

Severity and message	Cause	Action
Error: The default value(s) of a non-nullable personalization key should not be set to '<NULL>': '<PK Name>'.	A null default value has been set for a non-nullable personalization key. No error displays for this error.	Set the default value for the personalization key to a non-null value.
Warning: Default value for the '<Type>' data type for personalization key '<PK Name>' contains warning : '<Msg>'.	A structure personalization key contains a default value that generates a warning message.	Correct the problem identified by the warning message.
Error: Invalid default value '<Default Value>' for the '<Type>' data type of personalization key: '<PK Name>'.	The default value for the datatype of the personalization key is invalid.	Set the default value to conform to the datatype.
Error: Default value exceeds allowed maximum length: '<PK_length>'.	The length of the default value exceeds the datatype's length.	Set the default value to conform to the datatype length.

Custom Filters, Roles, and Synchronization Group Validation Rules and Error Messages

Validation rules are enforced during MBO development and, in some cases, when you deploy the MBO to Unwired Server. Messages are generated when validation rules are violated.

Table 16. Custom filter, role, and synchronization group validation rules and error messages

Severity and message	Cause	Action
Error: '<Filter class path>' has compile errors	A compilation error occurred in the result-set filter class file.	Correct the error in the filter class.
Warning: Duplicated result set filters in '<MBO Name>'. The filter class path '<Filter class path>' is shared by more than one filter.	One result-set filter is added to the same MBO more than once.	Remove duplicate filters.
Error: '<Filter class path>' does not implement <Interface name> interface.	The filter class does not implement the required interface.	Change the filter class to implement the required interface.

Severity and message	Cause	Action
Error: '<Filter class path>' does not exist in workspace.	The filter class has been deleted outside of the workspace, and the related result filter class information has not been updated to reference the MBO.	Delete the result set filter from the MBO, or add the original filter class back to the workspace.
Error: Duplicate case insensitive synchronization group name: '<SyncGroup Name>'.	The synchronization group name is a duplicate of another synchronization group.	Enter a valid and unique synchronization group name.
Error: The change detection interval must be between 1 and 2147483647 seconds for the synchronization group: '<SyncGroup Name>'.	The synchronization group interval has an invalid value.	Enter a valid interval value.
Warning: Relationship from 'MBO1' to 'MBO2' references two Mobile business Objects in different synchronization Groups. Mobile business objects with relationship should be grouped in the same synchronization group.	MBOs in a relationship are grouped in different synchronization groups.	Resolve MBO dependencies and group MBOs with relationships in the same synchronization group.
Error: Synchronization group name '<SyncGroup Name>' is a reserved name.	The synchronization group uses the reserved name system or initialSync.	Rename the synchronization group.
Error: Duplicate role name: '<Role Name>'.	The role name is a duplicate of an existing role.	Enter a unique and valid role name.

Mobile Workflow Validation Rules and Error Messages

Validation rules are enforced when defining mobile workflows. Messages are generated when validation rules are violated.

Table 17. Mobile workflow validation rules and error messages

Severity and message	Cause	Action
problem.empty.module.name=Module name cannot be empty	The Flow Design page property Module name requires an entry, but the property is blank.	Provide a module name.

Severity and message	Cause	Action
problem.module.name.too.long=The length of the module name cannot exceed 100 characters	The Module name is too long.	Shorten the module name.
problem.module.version.too.low=The module version must be at least 1	The Flow Design page property Module version requires an entry, but the property is blank.	Enter a module version of at least 1.
problem.match.rule.invalid.regex=The regular expression "{0}" for the "{1}" match rule is not valid.	The regular expression for the match rule indicated in the message is not valid.	Supply a regular expression that is a valid.
problem.screen.key.duplicated=The screen key "{0}" has been duplicated.	The screen key indicated in the messages is already used. The screen key value must be unique.	Enter another screen key that is unique.
problem.empty.screen.name=Screen name cannot be empty.	The Screen name property requires an entry, but the property is blank.	Enter a screen name.
problem.empty.screen.key=Screen key cannot be empty.	The Screen key property requires an entry, but the property is blank.	Enter a screen key.
problem.control.has.no.key=The control "{0}" on screen "{1}" has no key.	The control for the screen identified in the message requires a key, but the property is blank.	Enter a key for the control.
problem.control.uses.key.with.unsupported.datatype=The control "{0}" on screen "{1}" uses a key with a data type that this type of control does not support.	The data type entered for the control on the screen indicated in the message is not supported.	Enter a data type that is valid for the control.
problem.control.uses.key.used.by.another.control=The control "{0}" on screen "{1}" uses a key that another control on the same screen is using.	The key for the control and screen indicated in the message is already used by another control. The key must be unique.	Enter another control key that is unique.

Validation Rules

Severity and message	Cause	Action
problem.control.is.choice.but.has.no.options=The choice control "{0}" on screen "{1}" has no options.	The choice control requires one or more selection items. No items are included.	If you intend to use the choice control, select items to add from the drop-down list. Note that you can supply the selection items either statically or dynamically, but you must supply them. Or, select another control type.
problem.control.is.attachmentviewer.but.has.no.mimetype.or.filename=The attachment viewer control "{0}" on screen "{1}" has neither its MIME type nor its file name properties configured.	The attachment viewer control and screen named in the message does not include an attachment.	Configure either a MIME type or a file name to identify the attachment.
problem.too.many.default.menu.items=The screen "{0}" has more than one default menu items.	The screen identified in the message has multiple default menu items, but only one is allowed.	Identify only one menu item as the default.
problem.open.menu.item.with.no.screen=The menuitem "{0}" on the screen "{1}" is marked as being of type Open but does not have its screen property set.	The menu item on the screen identified in the message does not have screen properties set to correspond with the Open type.	Provide the screen to which to link.
problem.empty.flow.starting.point.name=Flow starting points cannot have an empty name.	The flow design requires starting point coordinates, but none are assigned.	Provide the starting point properties.
problem.operation.parameter.unmapped=No key is mapped to parameter {0}->{1}()->{2}	A key is required for the operation parameter indicated in the message.	Provide a key for the operation parameter.
problem.namedquery.parameter.unmapped=No key is mapped to parameter {0}->{1}()->{2}	A key is required for the named query parameter indicated in the message.	Provide a key for the named query parameter.
problem.key.unmapped=The key "{0}" is unused.	The key identified in the message is not used.	Either delete the key or map it to the required parameter.

Severity and message	Cause	Action
problem.screen.has.no.flow.starting.point=The screen "{0}" is not linked to any starting points and thus will be unused.	The screen identified in the message is not linked to any starting parts so is not used.	Either delete the screen or map it to the required starting points.
problem.action.has.pk.mapping.that.is.bound.to.nonexistent.pk=The screen "{0}" has a menu item "{1}" that has a personalization key mapping that refers to a personalization key, "{2}", that does not exist.	The personalization key that is identified in the messages is not mapped correctly.	Map to an existing personalization key.
problem.key.has.no.binding=The key "{0}" in the starting point "{1}" is not bound to any data.	The key identified in the message is not bound to a data source.	Map the starting point key to the data source.
problem.key.is.bound.to.nonexistent.pk=The key "{0}" in the starting point "{1}" is bound to a personalization key, "{2}", that does not exist.	The starting point key named in the message is bound to a personalization key that does not exist.	Either bind the key to an existing personalization key, or create the missing personalization key.
problem.key.in.non.server.initiated.starting.point.is.sent.from.server=The key "{0}" in the starting point "{1}", which is not server-initiated, is marked as being sent from the server, which is not supported.	The starting point key is not server-initiated, but it is associated with a server. Server-initiated starting point is not supported.	Change the starting point key to be non-server initiated.
problem.add.action.must.specify.a.screen=The menu item named "{0}" of type add on the screen "{1}" must specify a screen	The menu item on the screen identified in the message requires a screen, but one is not specified.	Specify a screen.
problem.key.name.duplicated=The starting point "{0}" has more than one key with the name "{1}"	The starting point named in the message has more than one key identified with the same name. A unique name is required for each key.	Rename any duplicate keys so all key names are unique.
problem.mismatched.match.rule.cc=The starting point "{0}"'s CC property does not match the workflow application's CC match rule.	The starting point named in the message for the e-mail header CC property does not match the CC match rule.	Check the starting point CC rule against the workflow application CC match rule, and make corrections so they match.

Validation Rules

Severity and message	Cause	Action
problem.mismatch-ed.match.rule.from=The starting point "{0}"s From property does not match the workflow application's From match rule.	The starting point named in the message for the From property does not match the From match rule.	Check the starting point From rule against the workflow application From match rule, and make corrections so they match.
problem.mismatch-ed.match.rule.subject=The starting point "{0}"s Subject property does not match the workflow application's Subject match rule.	The starting point named in the message for the Subject property does not match the Subject match rule.	Check the starting point Subject rule against the workflow application Subject match rule, and make corrections so they match.
problem.mismatch-ed.match.rule.to=The starting point "{0}"s To property does not match the workflow application's To match rule.	The starting point named in the message for the To property does not match the To match rule.	Check the starting point To rule against the workflow application To match rule, and make corrections so they match.
problem.credential.request.missing.username.key=The Credential Request starting point does not have a Credential Cache Username key	The starting point for Credential Request does not have a Credential Cache Username key assigned, and one is required.	If credentials are required, assign a Credential Cache Username key.
problem.credential.request.too.many.username.keys=The Credential Request starting point has too many Credential Cache Username keys	The starting point for Credential Request has more than one Credential Cache Username keys assigned, and only one is required.	Identify one Credential Cache Username key, and remove all others.
problem.credential.request.missing.password.key=The Credential Request starting point does not have a Credential Cache Password key	The starting point for Credential Request requires a Credential Cache Password key.	Specify a Credential Cache Password key.
problem.credential.request.too.many.password.keys=The Credential Request starting point has too many Credential Cache Password keys	The starting point for Credential Request has more than one Credential Cache Password keys assigned, and only one is required.	Identify one Credential Cache Password key, and remove all others.

Severity and message	Cause	Action
problem.missing.credential.cache.username.or.password=Please specify which editboxes are the credential cache username and password.	The EditBoxes for cache username and password are not clearly identified.	Identify the cache username and password EditBoxes clearly by setting the appropriate property in the property sheet of the username and password edit boxes.
problem.credential.request.screen.invokes.mbo=The screen pointed to by the Credential Request starting point should be used only to set the credential request user name and password.	The Credential Request starting point is set to an MBO. It should be set to a login screen.	Identify a login screen for entering log in credentials.
problem.credentials.are.not.cached.for.email.triggered.workflows=The credentials used in an notification triggered workflow are not cached between invocations. Consider using static credentials for notification triggered workflows.	You have specified dynamic credentials for the workflow, but static credentials may be more appropriate.	Consider using static credentials, where all users use the same login information, for notification triggered workflows.
problem.missing.critical.screen.flow=At least one Server-initiated or Client-initiated flow starting point needs to have a screen flow.	The screen flow design must include at least one server- or client-initiated flow starting point. One is not defined.	Include a server- or client-initiated flow starting point.
problem.operation.noinstance=The screen "{0}" has a menu item "{1}" which has one or more parameters that are either unmapped or are mapped to attributes of the "{2}" MBO without having an instance of that MBO.	The screen and menu item named in the message is not complete. Parameters are either unmapped or incorrectly mapped to an MBO.	Complete the menu mapping task.
problem.namedquery.noinstance=The screen "{0}" has a menu item "{1}" which is trying to invoke the object query "{2}" on the "{3}" MBO without having an instance of that MBO.	The screen and menu item named in the message cannot invoke an object query because the MBO instance is not included.	Determine why the MBO instance is not included, then fix the problem.

Validation Rules

Severity and message	Cause	Action
problem.listview.operation.no.params.bound.to.attributes=The screen "{0}" has a menu item "{1}" which is invoking an operation "{2}" and hasn't bound any of its parameters to attribute keys, which is not recommended.	The screen and menu item named cannot invoke a listview operation. Parameters are not bound to attribute keys.	Bind parameters to attribute keys, then retry the operation.
problem.no.credential.request.screen=This application, as designed, needs a Credential Request screen and does not have one.	The workflow application requires a Credential Request screen.	Add a Credential Request screen.
problem.no.credential.request.starting.point=This application, as designed, needs a Credential Request starting point	The workflow application requires a Credential Request starting point.	Add a Credential Request starting point.
problem.key.bound.to.relationship.that.is.many.to.one=The key "{0}" is bound to a relationship that is marked as being many-to-one. Many-to-one relationships are not supported.	The key is bound to a many-to-one relationship, which is not supported for mobile workflow applications. Only composite relationships are supported.	Change the key to bind to a composite relationship.
problem.key.bound.to.non.composite.relationship=The key "{0}" is bound to a relationship that is not marked as being composite. Only composite relationships are supported.	The key is bound to a relationship other than composite. Only composite relationships are supported.	Change the key to bind to a composite relationship.
problem.key.bound.invalid.relationship=The key "{0}" is bound to a relationship, "{1}", on the MBO "{2}", but that relationship no longer exists.	The key is bound to an MBO relationship that no longer exists.	Identify an MBO relationship that exists.
problem.first.screen.in.flow.has.save=The screen "{0}", the first screen in the "{1}" flow, has a menuitem, "{2}", which is of type Save. Save menuitems are not supported on the first screen of the flow.	The screen named and the first screen in the flow identified have the same Save menuitem. Save menuitems are not supported on the first screen of the flow.	Change the design so that Save menuitems are not used on the first screen of the flow. Consider using a Cancel, Close, or Submit workflow instead.

Severity and message	Cause	Action
problem.first.screen.in.flow.okaction.is.save=The screen "{0}", the first screen in the "{1}" flow, has its okaction property set to Save. Save actions are not supported on the first screen of the flow.	The named screen and first screen in the flow both have the okaction property set to Save. Save actions are not supported on the first screen of the flow.	Change the design so that the okaction property is set to Cancel on the first screen of the flow.
problem.mbo.not.found=The MBO "{0}" referenced from "{1}" could not be found	The MBO named in the reference could not be found.	Check the design to make sure the MBO reference is correct. You can use the MBO References dialog to retarget existing unfound MBO references to existing MBOs.
problem.mbo.not.found.on.email.starting.point=The MBO "{0}" referenced from the notification trigger could not be found	The MBO named in the reference could not be found.	Check the design to make sure the notification trigger reference is correct. You can use the MBO References dialog to retarget existing unfound MBO references to existing MBOs.
problem.key.mbo.attribute.not.found=The key "{0}" references a non-existent attribute, "{2}", of the MBO "{1}"	The key refers to an attribute that does not exist for the MBO named in the message.	Correct the key to refer to the correct attribute, or correct MBO with the attribute.
problem.fs.key.is.also.cc.key=The key of the editbox "{0}" from the screen "{1}" is a cached credential and flow-specific key.	Information only. The key identified with the editbox is a cached credential and flow-specific key.	None.
problem.default.value.is.not.numeric=The default value "{0}" for the control "{1}" on the screen "{2}" is an invalid numeric value.	The default value specified for the control on the screen is an invalid numeric value.	Change the default to a valid numeric value.
problem.key.type.does.not.match.option.type=The key type of "{0}" is not compatible with the control's logical type setting of "{1}" for the control "{2}" on the screen "{3}".	The key type specified is not valid for the control's logical type setting.	Specify a valid key type that is compatible with the control's logical type setting.

Validation Rules

Severity and message	Cause	Action
problem.menuitem.action.cant.be.asynchronous=The screen "{0}" has a menu item "{1}" which is of type Submit Workflow under conditions in which only online requests are supported.	This menu item cannot be bound to the "Submit Workflow" action type under the current design condition.	Change the design to allow use of the "Submit Workflow" action type for this menuitem, or use the "Online Request" action type for the menuitem.
problem.menuitem.duplicate.name=The screen "{0}" has two or more menu items with the same name, "{1}".	The screen named has duplicate menu items.	Give both menu items unique names.
problem.add.menuitem.empty.screen=The screen "{0}" has a menu item "{1}" which is of type Open and doesn't have a valid screen property setting.	A menu item on the named screen does not have valid screen property settings for the Open type.	Provide the correct property setting.
problem.add.menuitem.empty.screen=The screen "{0}" has a menu item "{1}" which is of type Add and doesn't have a valid screen property setting.	A menu item on the named screen does not have valid screen property settings for the Add type.	Provide the correct property setting.
problem.choice.control.refers.to.key.that.isnt.accessible=The choice control "{0}" on screen "{1}" refers to a key, "{2}", which it doesn't have access to.	The choice control on the named screen refers to a key that it does not have access.	Either change the control to refer to an accessible key, or arrange to have access to the key.

Glossary: Sybase Unwired Platform

Defines terms for all Sybase Unwired Platform components.

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

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

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

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

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

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

APNS – Apple Push Notification Service.

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

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

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

cache – The virtual tables in the consolidated database that store synchronization data. *See* CDB.

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

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

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

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

client application – *See* mobile application.

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

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

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

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

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

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

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

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

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

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

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

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

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

deployment archive – In Unwired WorkSpace, a deployment archive is created when a developer creates a package profile and executes the **build** operation. Building creates an archive that contains both a deployment unit and a corresponding descriptor file. A deployment archive can be delivered to an administrator for deployment to a production version of Unwired Server.

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

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

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

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

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

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

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

device user – The user identity tied to a device.

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

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

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

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

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

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

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

hostability – *See* multitenancy.

IDE – Integrated Development Environment.

JDE – BlackBerry Java Development Environment.

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

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

LDAP – Lightweight Directory Access Protocol.

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

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

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

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

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

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

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

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

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

Define MBO relationships to facilitate:

- Data synchronization
- EIS data-refresh policy

messaging based synchronization (MBS) – A synchronization method where data is delivered asynchronously using a secure, reliable messaging protocol. MBS provides fine-

grained synchronization (synchronization is provided at the data level—each process communicates only with the process it depends on), and it is therefore assumed that the device is always connected and available. *See also* replication based synchronization.

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

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

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

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

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

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

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

monitoring profiles – Monitoring profiles specify a monitoring schedule for a particular group of packages. These profiles let administrators collect granular data on which to base domain maintenance and configuration decisions.

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

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

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

openDS – The default LDAP server that is installed in Developer Edition and is suitable for authentication and authorization in a development environment.

operation – *See* MBO operation.

package – A package is a named container for one or more MBOs. On Unwired Server a package contains MBOs that have been deployed to this instance of the server.

palette – In Unwired WorkSpace, the palette is the graphical interface view from which you can add MBOs, local business objects, structures, relationships, attributes, and operations to the Mobile Application Diagram.

parameter – A parameter is a value that is passed to an operation/method. The operation uses the value to determine the output. When you create an MBO, you can map MBO parameters to data-source arguments. For example, if a data source looks up population based on a state abbreviation, the MBO gets the state from the user, then passes it (as a parameter) to the data source to retrieve the information. Parameters can be:

- Synchronization parameters – synchronize a device application based on the value of the parameter.
- Load parameters – perform a data refresh based on the value of the parameter.
- Operation parameters – MBO operations contain parameters that map to data source arguments. Operation parameters determine how the client passes information to the enterprise information system (EIS).

personalization key – A personalization key allows a mobile device user to specify attribute values that are used as parameters for selecting data from a data source. Personalization keys are also used as operation parameters. Personalization keys are set at the package level. There are three type of personalization keys: Session, client, server.

They are most useful when they are used in multiple places within a mobile application, or in multiple mobile applications on the same server. Personalization keys may include attributes such as name, address, zip code, currency, location, customer list, and so forth.

physical role – A security provider group or role that is used to control access to Unwired Server resources.

Problems view – In Eclipse, the Problems view displays errors – warnings for the Mobile Application Project.

provisioning – The process of setting up a mobile device with required runtimes and device applications. Depending on the synchronization model used and depending on whether or not the device is also an Afaria client, the files and data required to provision the device varies.

pull synchronization – Pull synchronization is initiated by a remote client to synchronize the local database with the CDB. On Windows Mobile, pull synchronization is supported only in RBS applications.

push synchronization – Push is the server-initiated process of downloading data from Unwired Server to a remote client, at defined intervals, or based upon the occurrence of an event.

queue – In-flight messages for a messaging application are saved in a queue. A queue is a list of pending activities. The server then sends messages to specific destinations in the order that they appear in the queue. The depth of the queue indicates how many messages are waiting to be delivered.

relationship – *See* MBO relationship.

relay server – *See also* Sybase Hosted Relay Service.

replication based synchronization (RBS) – A synchronization method where data is delivered synchronously using an upload/download pattern. For push-enabled clients, RBS uses a "poke-pull" synchronization model, where a notification is pushed to the device (poke), and the device fetches the content (pull), and is assumed that the device is not always connected to the network and can operate in a disconnected mode and still be productive. For clients that are not push-enabled, the default synchronization model is pull. *See also* messaging based synchronization.

REST web services – Representational State Transfer (REST) is a style of software architecture for distributed hypermedia systems such as the World Wide Web.

RFC – Remote Function Call. You can use the RFC interface to write applications that communicate with SAP R/3 applications and databases. An RFC is a standalone function. Developers use SAP tools to write the Advanced Business Application Programming (ABAP) code that implements the logic of a function, and then mark it as "remotely callable," which turns an ABAP function into an RFC.

role – Roles control access to Sybase Unwired Platform resources. *See also* logical role and physical role.

role mapping – Maps a physical (server role) to a logical (Unwired Platform role). Role mappings can be defined by developers, when they deploy an MBO package to a development Unwired Server, or by platform or domain administrators when they assign a security configuration to a domain or deploy a package to a production Unwired Server (and thereby override the domain-wide settings in the security configuration).

RSOE – Relay Server Outbound Enabler. An RSOE is an application that manages communication between Unwired Server and a relay server.

runtime server – An instance of Unwired Server that is running. Typically, a reference to the runtime server implies a connection to it.

SAP – SAP is one of the EIS types that Unwired Platform supports.

SCC – Sybase Control Center. A Web-based interface that allows you to administer your installed Sybase products.

scheduled data refresh – Data is updated in the consolidated database from a back-end EIS, based on a scheduled data refresh. Typically, data is retrieved from an EIS (for example, SAP) when a device user synchronizes. However, if an administrator wants the data to be preloaded for a mobile business object, a data refresh can be scheduled so that data is saved locally in a cache. By preloading data with a scheduled refresh, the data is available in the information server when a user synchronizes data from a device. Scheduled data refresh requires that an administrator define a cache group as "scheduled" (as opposed to "on-demand").

security configuration – Part of the application user and administration user security. A security configuration determines the scope of user identity, authentication and authorization checks, and can be assigned to one or more domains by the platform administrator in Sybase Control Center. A security configuration contains:

- A set of configured security providers (for example LDAP) to which authentication, authorization, attribution is delegated.
- Role mappings (which can be specified at the domain or package level)

security provider – A security provider and its repository holds information about the users, security roles, security policies, and credentials used by some to provide security services to Unwired Platform. A security provider is part of a security configuration.

security profile – Part of the Unwired Server runtime component security. A security profile includes encryption metadata to capture certificate alias and the type of authentication used by server components. By using a security profile, the administrator creates a secured port over which components communicate.

server connection – The connection between Unwired WorkSpace and a back-end EIS is called a server connection.

server farm – *See also* cluster. Is the relay server designation for a cluster.

server-initiated synchronization – *See* push synchronization.

SOAP – Simple Object Access Protocol. SOAP is an XML-based protocol that enables applications to exchange information over HTTP. SOAP is used when Unwired Server communicates with a Web service.

solution – In Visual Studio, a solution is the high-level local workspace that contains the projects users create.

Solution Explorer – In Visual Studio, the Solution Explorer pane displays the active projects in a tree view.

SSO – Single sign-on. SSO is a credential-based authentication mechanism.

statistics – In Unwired Platform, the information collected by the monitoring database to determine if your system is running as efficiently as possible. Statistics can be current or historical. Current or historical data can be used to determine system availability or performance. Performance statistics are known as key performance indicators (KPI).

Start Page – In Visual Studio, the Start Page is the first page that displays when you launch the application.

structured data – Structured data can be displayed in a table with columns and labels.

structure object – Defined in Unwired WorkSpace, structures hold complex datatypes, for example, a table input to a SAP operation.

subscription – A subscription defines how data is transferred between a user's mobile device and Unwired Server. Subscriptions are used to notify a device user of data changes, then these updates are pushed to the user's mobile device.

Sybase Control Center – Sybase Control Center is the Flash-based Web application that includes a management framework for multiple Sybase server products, including Unwired Platform. Using the Unwired Platform administration perspective in Sybase Control Center, you can register clusters to manage Unwired Server, manage domains security configurations, users, devices, connections and monitor the environment. You can also deploy MBO packages and manage deployed MBO packages in order to design the synchronization behavior for those packages. Only use the features and documentation for Unwired Platform. Default features and documentation in Sybase Control Center do not always apply to the Unwired Platform use case.

Sybase Hosted Relay Service – The Sybase Hosted Relay Service is a Web-hosted relay server that enables you to test your Unwired Platform development system.

Sybase Messaging Service – The synchronization service that facilitates communication with device client applications.

Sybase Unified Agent – Provides runtime services to manage, monitor, and control distributed Sybase resources. The agent must be running for Sybase Control Center to run.

Sybase Unwired Platform – Sybase Unwired Platform is a development and administrative platform that enables you to mobilize your enterprise. With Unwired Platform, you can develop mobile business objects in the Unwired WorkSpace development environment, connect to structured and unstructured data sources, develop mobile applications, deploy mobile business objects and applications to Unwired Server, which manages messaging and data services between your data sources and your mobile devices.

Sybase Unwired WorkSpace – Sybase Unwired Platform includes Unwired WorkSpace, which is a development tool for creating mobile business objects and mobile applications.

synchronization group – Defined in Unwired WorkSpace, a synchronization group is a collection of MBOs that are synchronized at the same time.

synchronization parameter – A synchronization parameter is an MBO attribute used to filter and synchronize data between a mobile device and Unwired Server.

synchronization phase – For replication based synchronization packages, the phase can be an upload event (from device to the consolidated database) or download event (from the consolidated database to the device).

synchronize – *See also* data refresh. Synchronization is the process by which data consistency and population is achieved between remote disconnected clients and Unwired Server.

truststore – The location in which certificate authority (CA) signing certificates are stored. *See also* keystore.

undeploy – Running **undeploy** removes a domain package from an Unwired Server.

Unwired Server – The application server included with the Sybase Unwired Platform product that manages mobile applications, back-end EIS synchronization, communication, security, transactions, and scheduling.

user – Sybase Control Center displays the mobile-device users who are registered with the server.

Visual SQL – A graphical user interface tool that you can use to build SQL queries.

Visual Studio – Microsoft Visual Studio is an integrated development environment product that you can use to develop device applications from generated Unwired WorkSpace code.

Welcome page – In Eclipse, the first set of pages that display when you launch the application.

workspace – In Eclipse, a workspace is the directory on your local machine where Eclipse stores the projects that you create.

WorkSpace Navigator – In Eclipse, the tree view that displays your mobile application projects.

WSDL file – Web Service Definition Language file. The file that describes the Web service interface that allows clients to communicate with the Web service. When you create a Web service connection for a mobile business object, you enter the location of a WSDL file in the URL.

Index

A

- administration performance 66
- administration perspective
 - empty SCC console screen 67
 - gray SCC console screen 67
 - rectangular box instead of SCC console 67
- administrators
 - login accepted when authentication provider unavailable 72
- Advantage Database Server 63
- agent.log file 68, 76
- authentication
 - provider unavailable but administrator can log in 72
- AuthorizationException 80
- auto purge
 - removing monitor data 73

C

- caching of login credentials 80
- CallbackHandler 28, 30
- cannot access Device tab 45
- cannot access sampledb
 - deployment (production) edition 87
 - development edition 87
- cannot authenticate user 63
- cannot generate code with Javadoc option enabled 101
- cannot insert DEFAULT in Timestamp column 105
- cannot start Unwired Server 76
- cluster problems 63
- code generation fails 101
- components
 - Windows services reference 43
- connection errors 78
- connection pools
 - configuring 54
- connections
 - troubleshooting 50, 54
- consolidated database
 - troubleshooting 54
- consolidated database cache problems 51
- credentials
 - old, ability to authenticate with 80

D

- data source
 - connections fail 88
 - troubleshooting 50
- debugging 26, 28, 30
 - SAP DOE-C process flow 33
 - SAP JCO process flow 34
- DEFAULT in Timestamp column 105
- degrading performance 81
- deploy failure for large packages 81
- deployment issues for packages 81
- deployment problems 49
- device log error 83
- device management issues 83
- device provisioning problems 113
- device user name not displayed 84
- DNS server failure 69
- documentation roadmap
 - document descriptions 1
- DOE-C
 - connection test errors 54
 - invalid user 82
 - SAP package fails 54
- DOE-C process flow 33
- domain logs 8

E

- EIS connection problems 54
- ensuring system recoverability 13
- error messages
 - described 5
 - mobile business objects 125
 - mobile workflow applications 125, 154
 - severity levels 6
 - variables 6
- errors
 - user account failure 82
- exception:java.lang.Error 119
- export failure for large packages 81

F

- F5 (browser refresh)
 - logging out of Sybase Control Center 71
- Failover Clustering 63

G

- glossaries
 - Sybase Unwired Platform terms 163
- graphic image causes OutOfMemory error 120

H

- History tab is blank 73
- host name changes not reflected in SCC 72
- host name resolution failure 69
- hosted Relay Server problems 60, 61

I

- intrusion detection/prevention software 48
- invalid login 70

J

- JCO process flow 34
- JMSBridge 28, 30

L

- LDAP
 - processes 70
 - startup 70
- listener startup errors 78
- log files
 - agent.log file 68
 - described 7
 - location 9
 - scc_repository.log 68
- logging in to Sybase Control Center
 - clearing authentication parameters 71
- logging out of Sybase Control Center
 - unintentionally, using F5 browser refresh 71
- login invalid 70
- login troubleshooting
 - Sybase Control Center 71
- logs
 - domain-level 8

M

- management console unavailable 72
- management issues for packages 81
- maxPoolSize 54
- maxWaitTime 54
- MBOLogger 26, 28, 30
- messaging
 - connection test errors 54
- messaging queue counts 81
- Messaging Synchronization Server
 - problems 62
- Microsoft Cluster Service 63
- mobile business objects
 - error messages 125
 - validation rules 125
- mobile workflow applications
 - error messages 125, 154
 - validation rules 125, 154
- monitoring
 - issues for platform components 72
- monitoring data
 - not displayed 73
- MS Cluster issues 72

N

- NetworkStreamParams property 114

O

- OutOfMemory error on BlackBerry simulator 120

P

- package deployment and management issues 81
- package type mismatch 123
- packet dropped error 116
- passwords
 - old, ability to authenticate with 80
- performance data
 - administration 66
- performance degradation 81
- platform component monitoring issues 72
- port conflict
 - cannot access Device tab 45
- port conflicts
 - among multiple SCC versions 70

- with Sybase Unified Agent 78
- port numbers 38, 45, 70
- problems starting Sybase Unified Agent services 68
- problems with device and device user management 83

Q

- queue counts 81

R

- reinstalling Unified Agent 69
- Relay Server
 - connection failure 61
 - problems 60
- Relay Server problems 61
- remote server environment
 - server error 113
 - synchronization failure 113
- restarting a remote server
 - unsuccessful 77
- RSOE
 - startup problems 48

S

- sampledb configuration error 113
- SAP 56, 64
 - debugging DOE-C process flow 33
 - debugging JCO process flow 34
 - DOE-C connection test errors 54
 - messaging connection test errors 54
 - user account error 82
- SCC console tree is not complete 80
- scc_repository.log file 68
- security error when connecting to SCC 72
- security problems 52
- server problems 57
- Server Response Timeout error on BlackBerry
 - device 121
- Server Response Timeout error on BlackBerry
 - simulator 121
- server tier administration issues 74
- ServerAliases 63
- servers
 - problems 57, 60, 62
- services, Windows 43

- severity levels in error messages 6
- SSL certificates
 - error when missing 72
- starting a remote server
 - unsuccessful 77
- startup problems
 - RSOE 48
 - Unwired Server 48
- stopping a remote server
 - unsuccessful 77
- stream parameters 114
- SUPBridge 28, 30
- Sybase Control Center
 - cannot ping Unwired Server 87
 - failure to start 69
 - logging out unintentionally with F5 71
 - management tier issues 67
 - second version fails to start 70
 - security error when connecting 72
 - Sybase Unwired WorkSpace
 - cannot ping Unwired Server 87
- Sybase hosted Relay Server problems 60, 61
- Sybase Unified Agent 70
 - port conflicts 78
 - Windows service fails to start 68
- synchronization fails 113
- synchronization problems 49, 76, 114
- system messages 5
- system recoverability 13

T

- TCP/IP filtering causing errors 78
- terms
 - Sybase Unwired Platform 163
- testing database file on SD card from Windows
 - Mobile simulator 124
- time-change related problems for Unwired Server 59
- troubleshooting
 - basic scenarios 15
 - basics 5
 - cache 22
 - changing host name 37
 - cluster problems 63
 - consolidated database cache problems 51
 - data change notification 23
 - deployment problems 49
 - device client problems 113
 - device provisioning 113

- Eclipse mobile business object development
 - 87
 - EIS connection problems 54
 - end-to-end process flow 15
 - message-based synchronization 17
 - Mobile Workflow Package Designer 107
 - post-installation configuration 37
 - Relay Server problems 60
 - replication-based synchronization 20
 - security problems 52
 - server problems 57
 - Sybase Control Center problems 65
 - Sybase hosted Relay Server 61
 - synchronization problems 49
 - system level problems 37
 - system performance issues 57
 - Unwired Server problems 57, 62, 76
 - Unwired Server startup 76
 - user account failure 82
 - troubleshooting MS Cluster 72
 - troubleshooting performance issues 66
 - troubleshooting Unwired Platform with SCC 65
- U**
- Unified Agent
 - reinstalling the service 69
 - Unwired Platform administration nodes
 - cannot access 71
 - Unwired Platform management console unavailable
 - 72
 - Unwired Server
 - cannot ping from Sybase Control Center 87
 - cannot ping from Sybase Unwired WorkSpace
 - 87
 - configuration changes unsuccessful 79
 - extended session 70
 - list does not appear in SCC 74
 - problems 57
 - refresh after changing configuration 79
 - startup failure 76
 - startup failure after time change 59
 - startup problems 48
 - user management issues for devices 83
 - users
 - able to connect with old password 80
 - not displayed for registered devices 84
- V**
- validation rules
 - mobile business objects 125
 - mobile workflow applications 125, 154
 - variables in error messages 6
- W**
- Windows
 - services reference 43
 - Windows Mobile devices
 - cannot test database file on SD card from
 - simulator 124