

IBM DB2 Everyplace



Sync Server Administration Guide

Version 9 Release 1

IBM DB2 Everyplace



Sync Server Administration Guide

Version 9 Release 1

Note

Before using this information and the product it supports, read the information in "Notices" on page 249.

| **Eighth Edition (July 2006)**

| This edition applies to Version 9, Release 1 of IBM DB2 Everyplace and to all subsequent releases and modifications
| until otherwise indicated in new editions.

© **Copyright International Business Machines Corporation 1998, 2006. All rights reserved.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this book vii

Conventions used in this book	vii
How to read syntax diagrams	vii
Service updates and support information	viii
Receiving information updates automatically	ix
The DB2 Everyplace information set	ix
Accessibility features	ix
How to send your comments	x

IBM DB2 Everyplace Sync Server Administration Guide version 9.1 1

Product overview 3

An example DB2 Everyplace scenario	3
Components of the DB2 Everyplace solution.	4
DB2 Everyplace environments	5
The DB2 Everyplace sample applications	7

The administration process 9

Managing users and data	9
Mobile Devices Administration Center.	9
What is the Mobile Devices Administration Center?	9
Starting the Mobile Devices Administration Center	10
The Mobile Devices Administration Center interface	10
Mobile Devices Administration Center and synchronization objects	11
What is a synchronization object?	12
Creating synchronization objects	13
Where synchronization objects are stored	14
Editing a synchronization object	14
Deleting an object	14
Effects of administrative actions on pending client changes	15
Information displayed in the contents pane.	16
Filtering the contents pane	17
Refreshing the view of an object folder	18
Administering user accounts	18
What is a user?	18
Evaluating the characteristics of your mobile users	18
Creating users in the Mobile Devices Administration Center.	19
Disabling synchronization	19
Resetting groups, users, and devices	20
Deleting groups, users, and devices	20
Allowing multiple devices per user	21
Creating mobile users	22
Concluding the administration process	23
Creating identities for users	23
Assigning a user to a group from the Users window	23
Setting passwords for users	23

Importing user definitions from another source	24
Administering groups of users	24
What is a data synchronization group?	24
Creating a data synchronization group	24
Identifying the group	25
Enabling the group to access data and files.	26
Assigning a user to a group from the Groups window	26
Enabling synchronization.	26
Administering subscriptions.	27
Data setup and replication	27
The default adapters	27
Administering JDBC subscriptions.	27
What is a JDBC subscription?	27
Specifying source tables, target tables, and privileges for the subscription	28
Specifying the mirror database driver and URL.	29
Specifying the source database driver and URL.	29
Configuring the DB2 Everyplace Sync Server to access data sources using JDBC	30
Specifying the encryption level of a JDBC subscription	31
Identifying the JDBC subscription	32
Creating a JDBC subscription	33
Restrictions for table subscriptions.	35
Administering upload subscriptions	36
What is an upload subscription?	36
Specifying the source database driver and URL.	37
Specifying source tables and target tables	38
Specifying the encryption level of an upload subscription	39
Creating an upload subscription	39
Identifying the upload subscription	40
Administering File subscriptions	40
What is a file subscription?	40
Specifying subscription sets that use the subscription	41
Flagging a subscription that references a changed file	42
Specifying the name of the source file	42
Identifying the file subscription.	43
Specifying the encryption level of a file subscription	44
Creating a file subscription	44
Administering Custom subscriptions	46
Administering Data Propagator subscriptions	46
What is a DataPropagator subscription?	46
Creating a DataPropagator table subscription	46
Data source restrictions for DataPropagator subscriptions	54
Administering subscription sets	55

What is a subscription set?	55	Planning and design	127
What is a replication subscription?	55	A comparison of the Data Propagator and JDBC replication engines	127
Grouping subscriptions into mobile subscription sets.	56	Overview of DB2 Everyplace subscriptions and replication engines	127
Enabling groups to access their data and files	57	Differences between the Data Propagator replication engine and the JDBC replication engine	128
Making the subscription set available to a group	57	Conflict resolution.	129
Specifying what information the subscription set accesses	58	The Data Propagator replication engine.	129
Identifying the subscription set.	58	How the Data Propagator replication engine handles data changes to the source system	130
Assigning a subscription set to a Custom subscription	59	How the Data Propagator replication engine handles data changes to the client system	131
Assigning a subscription set to an upload subscription	60	The JDBC replication engine	132
Assigning a subscription set to a JDBC subscription	60	How the JDBC replication engine handles data changes to the source system	132
Administering table filters in order to manage data	61	How the JDBC replication engine handles data changes to the client system.	133
What is a data filter?	62	Changing the batch size for the JDBC replication engine	134
Filtering data at the subscription level	62		
Filtering Data Vertically	63		
Filtering columns from the source database to the mirror database.	63		
Filtering columns from the mirror database to the client device	63		
Filtering data horizontally	66		
Filtering rows from the source database to the mirror database.	66		
Filtering rows from the mirror database to the client device	67		
Filter terms defined.	73		
Data filter example	75		
Filter parameter uses and limitations	76		
Using custom logic in DB2 Everyplace	76		
What is custom logic	76		
Custom Logic terminology	77		
Developing custom logic	77		
Deploying custom logic classes	79		
Using Custom Logic for conflict resolution	80		
Enabling custom logic policies for the sample application	81		
Using the custom logic sample application	82		
Data type mapping for Custom Logic	83		
Custom Logic API	84		
Using the XML scripting tool	84		
About the XML scripting tool	85		
Overview of the XML scripting tool	85		
Exporting XML scripts.	86		
Invoking the XML Scripting tool from the command line	86		
Invoking the XML tool programmatically	87		
Batching DB2 Everyplace Sync Server administration commands	90		
Sample XML scripts	90		
JDBC subscriptions	101		
DataPropagator subscriptions	106		
Creating a group or a user	108		
Creating subscription sets by using the XML Scripting tool	109		
XML Scripting tool restrictions.	110		
XML scripting tool DTD.	110		
		Synchronizing databases in DB2 Everyplace	135
		The synchronization process	135
		Enabling and disabling synchronization	135
		Synchronizing mobile devices	136
		Synchronization from the mobile device to the source database	136
		Synchronizing using DB2 Sync	138
		The DB2 Sync sample application	138
		Configuring DB2 Sync for a Palm OS device	138
		Configuring Server settings on DB2 Sync for a Palm OS device	139
		DB2 Sync menu options	139
		Synchronizing data using DB2 Sync	141
		Synchronization on the server	142
		Synchronization from the source database to the mobile device	142
		Synchronization tutorial	144
		Creating a JDBC subscription to access source data	144
		Creating a data synchronization group	146
		Starting the application server.	147
		DB2 Everyplace support and troubleshooting	155
		Testing the DB2 Everyplace Sync Server servlet	155
		Diagnostic data for the DB2 Everyplace mobile database	155
		Diagnostic data for the DB2 Everyplace Sync Server	156
		Enabling tracing for the DB2 Everyplace Sync Client	156
		Verifying database integrity with the data integrity check tool	157
		Handling DB2 Everyplace synchronization problems	157

Synchronization conflict resolution	157
The order of synchronization and reception of error messages	159
Viewing the error log to diagnose problems	159
Defining the tracing level	160
Viewing the log on the mobile device	161
Purging error log entries automatically	161
Providing error-handling logic for user-exits	161
Monitoring Synchronization in DB2 Everyplace	165
Monitoring DB2 Everyplace mobile device synchronization	165
Using the diagnosis utility	165
Using the recovery utility	165
Collecting synchronization information	166
Enabling the client synchronization statistics subscription	166
Enabling users to upload synchronization trace data	166
Setting synchronization statistics properties using the Mobile Devices Administration Center	166
Configuring synchronization information databases on the client	167
Configuring client synchronization information databases using the Java API	167
Configuring client synchronization information databases using the C API	168
Querying client synchronization information tables	169
Using the sample synchronization statistics report	169
Client synchronization statistics table structures	169
Monitoring DB2 Everyplace server synchronization	172
Specifying pruning options for the message store	172

DB2 Everyplace Sync Server replication statistics	173
Specifying pruning options for replication statistics	173

Reference for DB2 Everyplace 175

Command-line scripts for the DB2 Everyplace Sync Server	175
Data type mappings between DB2 Everyplace and data sources	184
Supported database default values	184
DB2 family data type mappings	186
Informix data type mappings	187
Oracle data type mappings	189
Microsoft SQL Server data type mappings	189
Data type mapping restrictions	190
Data source restrictions for DataPropagator subscriptions	191
DB2 Everyplace limits	192
Overview of the DB2 Everyplace mobile database tables	194
DB2 Everyplace System Catalog base tables	194
DB2 Everyplace error messages	196
Error messages	197
DB2 Everyplace Update Tool error messages	237
DB2eCLP	238
DB2eCLP commands	238
Importing and exporting data using the DB2eCLP	240

Glossary 243

Notices 249

Trademarks	251
----------------------	-----

Index 253

About this book

This book is designed to help database administrators, system programmers, application programmers, and system operators perform the following tasks:

- Use and maintain the DB2 Everyplace Sync Server
- Connect the DB2 Everyplace Sync Server to data sources
- Configure communications between the DB2 Everyplace Sync Server and mobile and embedded devices
- Manage users and data
- Diagnose and recover from DB2 Everyplace problems

Always check the DB2 Everyplace Library page for the most current version of this publication:

<http://www.ibm.com/software/data/db2/everyplace/library.html>

Conventions used in this book

This documentation uses the following highlighting conventions:

- **Boldface type** indicates commands or user interface controls such as names of fields, folder, icons, or menu choices.
- Monospace type indicates examples of text that you enter exactly as shown.
- *Italic type* indicates variables that you should replace with a value. It is also used to indicate book titles and to emphasize significant words.

In this documentation, <DSYPATH> refers to the directory where DB2 Everyplace is installed. For instructions that are specific to Linux and UNIX systems, \$DSYINSTDIR refers to the directory where the DB2® Everyplace® Sync Server instance is located for a given user ID.

How to read syntax diagrams

The following rules apply to the syntax diagrams that are used in this information:

- Read the syntax diagrams from left to right, from top to bottom, following the path of the line. The following conventions are used:
 - The >>--- symbol indicates the beginning of a syntax diagram.
 - The ---> symbol indicates that the syntax diagram is continued on the next line.
 - The >--- symbol indicates that a syntax diagram is continued from the previous line.
 - The --->< symbol indicates the end of a syntax diagram.
- Required items appear on the horizontal line (the main path).

▶▶—required_item—————▶▶

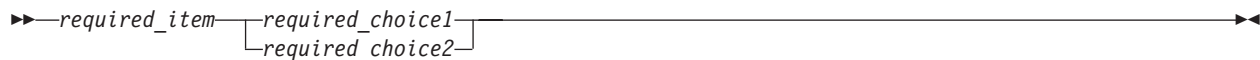
- Optional items appear below the main path.

▶▶—required_item—└ optional_item ┘—————▶▶

If an optional item appears above the main path, that item has no effect on the execution of the syntax element and is used only for readability.



- If you can choose from two or more items, they appear vertically, in a stack. If you *must* choose one of the items, one item of the stack appears on the main path.



If choosing one of the items is optional, the entire stack appears below the main path.



If one of the items is the default, it appears above the main path, and the remaining choices are shown below.



- An arrow returning to the left, above the main line, indicates an item that can be repeated.



If the repeat arrow contains a comma, you must separate repeated items with a comma.



A repeat arrow above a stack indicates that you can repeat the items in the stack.

- Keywords, and their minimum abbreviations if applicable, appear in uppercase. They must be spelled exactly as shown. Variables appear in all lowercase italic letters (for example, *column-name*). They represent user-supplied names or values.
- Separate keywords and parameters by at least one space if no intervening punctuation is shown in the diagram.
- Enter punctuation marks, parentheses, arithmetic operators, and other symbols, exactly as shown in the diagram.
- Footnotes are shown by a number in parentheses, for example (1).

Service updates and support information

To find service updates and support information, including software fix packs, Frequently Asked Questions (FAQs), technical notes, troubleshooting information, and downloads, refer to the following Web page:

<http://www.ibm.com/software/data/db2/everyplace/support.html>

Receiving information updates automatically

By registering with the IBM My Support service, you can automatically receive a weekly e-mail that notifies you when new DCF documents are released, when existing product documentation is updated, and when new product documentation is available. You can customize the service so that you receive information about only those IBM products that you specify.

To register with the My Support service:

1. Go to <http://www.ibm.com/support/mysupport>.
2. Enter your IBM ID and password, or create one by clicking **register now**.
3. When the My Support page is displayed, click **add products** to select those products that you want to receive information updates about. DB2 Everyplace is located under **Software -> Data and Information Management -> Databases**.
4. Click **Subscribe to email** to specify the types of updates that you would like to receive.
5. Click **Update** to save your profile.

The DB2 Everyplace information set

DB2 Everyplace technical information is available in the following formats:

PDF files

The PDF versions of the books are titled as follows:

- *DB2 Everyplace Application and Development Guide*
- *DB2 Everyplace Sync Server Administration Guide*
- *DB2 Everyplace Installation and User's Guide*

A description of each book in the DB2 Everyplace library is available from:

- The IBM® Publications Center at <http://www.ibm.com/shop/publications/order>
- The DB2 Everyplace Web site at <http://www.ibm.com/software/data/db2/everyplace/library.html>

IBM developerWorks

IBM developerWorks has technical articles about DB2 Everyplace and a forum for interacting with other DB2 Everyplace users. You can access the DB2 Everyplace section of IBM developerWorks at <http://www.ibm.com/developerworks/db2/products/db2e/index.html>.

Information center

The information center contains the entire DB2 Everyplace library in a convenient searchable format. More information is available on the DB2 Everyplace library page at <http://www.ibm.com/software/data/db2/everyplace/library.html>.

under the **Information Center** heading.

Online help

You can open HTML browser-based online help from the Mobile Devices Administration Center user interface.

Accessibility features

Accessibility features help users with physical disabilities, such as restricted mobility or limited vision, to use software products successfully.

The Installer, Configuration Wizard, and Mobile Devices Administration Center are all accessible and include the following accessibility features:

- Operate all features by using the keyboard instead of the mouse.
- Customize the size and color of your fonts.
- Receive either visual or audio alert cues.
- Supports accessibility applications that use the Java™ Accessibility API.
- Comes with documentation that is provided in an accessible format.

Keyboard input

You can use keys or key combinations to perform operations that can also be done by using a mouse. You can access context-sensitive menus from the menu bar instead of right-clicking.

Accessible display

DB2 Everyplace has features that enhance the user interface and improve accessibility for users with low vision. These accessibility enhancements include support for customizing font properties.

Font settings

You can select the color, size, and font for the text in menus and dialog windows.

No dependence on color

You do not need to distinguish between colors in order to use any of the functions in this product.

Supports high contrast colors

The Mobile Devices Administration Center displays well when you use a high contrast color scheme.

No flashing or blinking content

No text or graphical user interface elements flash or blink during operation.

Alternative alert cues

You can specify whether you want to receive alerts through audio or visual cues.

Compatibility with assistive technologies

The Mobile Device Administration Center interface supports the Java Accessibility API, enabling use by screen readers and other assistive technologies used by people with disabilities.

Accessible documentation

Accessible documentation for DB2 Everyplace is available in the DB2 Everyplace information center.

How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book or any other DB2 Everyplace documentation, use either of the following options:

- Use the online reader comment form, which is located at:
<http://www.ibm.com/software/data/rcf/>
- Send your comments by e-mail to comments@us.ibm.com. Be sure to include the name of the book, the part number of the book, the version of DB2 Everyplace, and, if applicable, the specific location of the text you are commenting on (for example, a page number or table number).

IBM DB2 Everyplace Sync Server Administration Guide version 9.1

Product overview

DB2 Everyplace is part of the IBM On Demand Business solution for synchronizing data between mobile devices and enterprise database servers.

By using DB2 Everyplace, mobile professionals (such as sales people, inspectors, auditors, field service technicians, doctors, realtors, and insurance claim adjusters) can have access to vital data that they need when they are away from the office. Organizations can deliver their enterprise data to mobile devices. By using the DB2 Everyplace mobile database, you can access and perform updates to databases that reside on mobile devices. By using the DB2 Everyplace Sync Server and the Sync Client, you can synchronize data from mobile devices to other data sources in your enterprise.

The following editions of DB2 Everyplace are available:

DB2 Everyplace Enterprise Edition

Provides database and synchronization software for mobile devices. This edition allows you to create a complete enterprise synchronization solution for your mobile devices. This edition includes:

- DB2 Everyplace Sync Server (including the Mobile Devices Administration Center and the XML Scripting tool)
- DB2 Everyplace mirror database
- DB2 Everyplace Sync Client
- DB2 Everyplace mobile database

DB2 Everyplace Database Edition

Provides the DB2 Everyplace mobile database. This edition is ideal if you only want a robust and small database for your mobile devices and do not need to synchronize data with an enterprise data source.

An example DB2 Everyplace scenario

DB2 Everyplace can increase the productivity and efficiency of a mobile workforce. In this example, an insurance adjuster uses a mobile device that runs a DB2 Everyplace application.

Insurance claims adjusters are responsible for inspecting the damaged property of customers who file claims. In many companies, the adjuster visits the claimant's property, fills out paper forms to validate or refute the claim, and assesses the amount of the damages to be paid to the claimant. Later, when the adjuster returns to the office, the forms are manually entered into the company's computer system in a tedious and expensive process.

Equipping the adjusters with a mobile device that runs a DB2 Everyplace application can considerably streamline this process. By using their mobile devices wherever they are, the adjusters can access their inspection schedule, route, and claimant policy information. The adjusters can also complete the adjustment form on the mobile device. The adjusters can then synchronize the data on their mobile devices with the company's computer system by uploading the new adjustment form data to the company's enterprise database. If the adjusters need information in the field, they can synchronize the data on their mobile devices with the company's computer system immediately by using modem or wireless connection.

The claims adjustment process can now be completely paper free, which translates into significant cost savings for the insurance company. Claims are also settled faster because adjusters have instant access to their company's enterprise databases.

Components of the DB2 Everyplace solution

DB2 Everyplace Enterprise Edition is a robust solution for synchronizing enterprise data. You can configure the DB2 Everyplace environment in multiple ways depending on the needs of your network and your users.

DB2 Everyplace mobile database

The DB2 Everyplace mobile database engine runs on a mobile device and stores a local copy of data from a source system. Users can use the mobile device to access and modify this data. The DB2 Everyplace mobile database is included with DB2 Everyplace Database Edition, DB2 Everyplace Enterprise Edition, and the Mobility on Demand feature of DB2.

The DB2 Everyplace mobile database is a relational database that resides on your mobile device. You can interact with the database through DB2 Call Level Interface (CLI) functions, Java Database Connectivity (JDBC) methods, Open Database Connectivity (ODBC) methods, or ADO.NET methods.

DB2 Everyplace Sync Server

The DB2 Everyplace Sync Server is a servlet that synchronizes data and provides conflict resolution between embedded databases on mobile devices and a source database. When you install DB2 Everyplace, you install the DB2 Everyplace Sync Server servlet and a limited functionality embedded application server. You can also configure the DB2 Everyplace Sync Server to run inside a stand-alone application server such as WebSphere Application Server version 6.

You can administer the DB2 Everyplace Sync Server by using two tools:

Mobile Devices Administration Center

This graphical tool helps you manage and deliver synchronization services to groups of users with similar data synchronization needs.

The XML Scripting tool

The XML Scripting tool automates tasks otherwise performed using the Mobile Devices Administration Center. You can also use the XML Scripting tool to copy or move subscriptions, subscription sets, users, groups from one server to several other servers.

DB2 Everyplace mirror database

The DB2 Everyplace mirror database stores the data that you want to synchronize between your mobile devices and your enterprise databases. The DB2 Everyplace Sync Server uses the mirror database to perform conflict resolution between mobile devices and to minimize load on your enterprise database systems.

If you have a stand-alone copy of DB2 Version 9.1 on your system, create or catalog the mirror database on the local DB2 instance of DB2 Everyplace. If you do not have DB2 Version 9.1 on your system, DB2 Everyplace installs an embedded, restricted version of DB2 Version 9.1 to function as the mirror database.

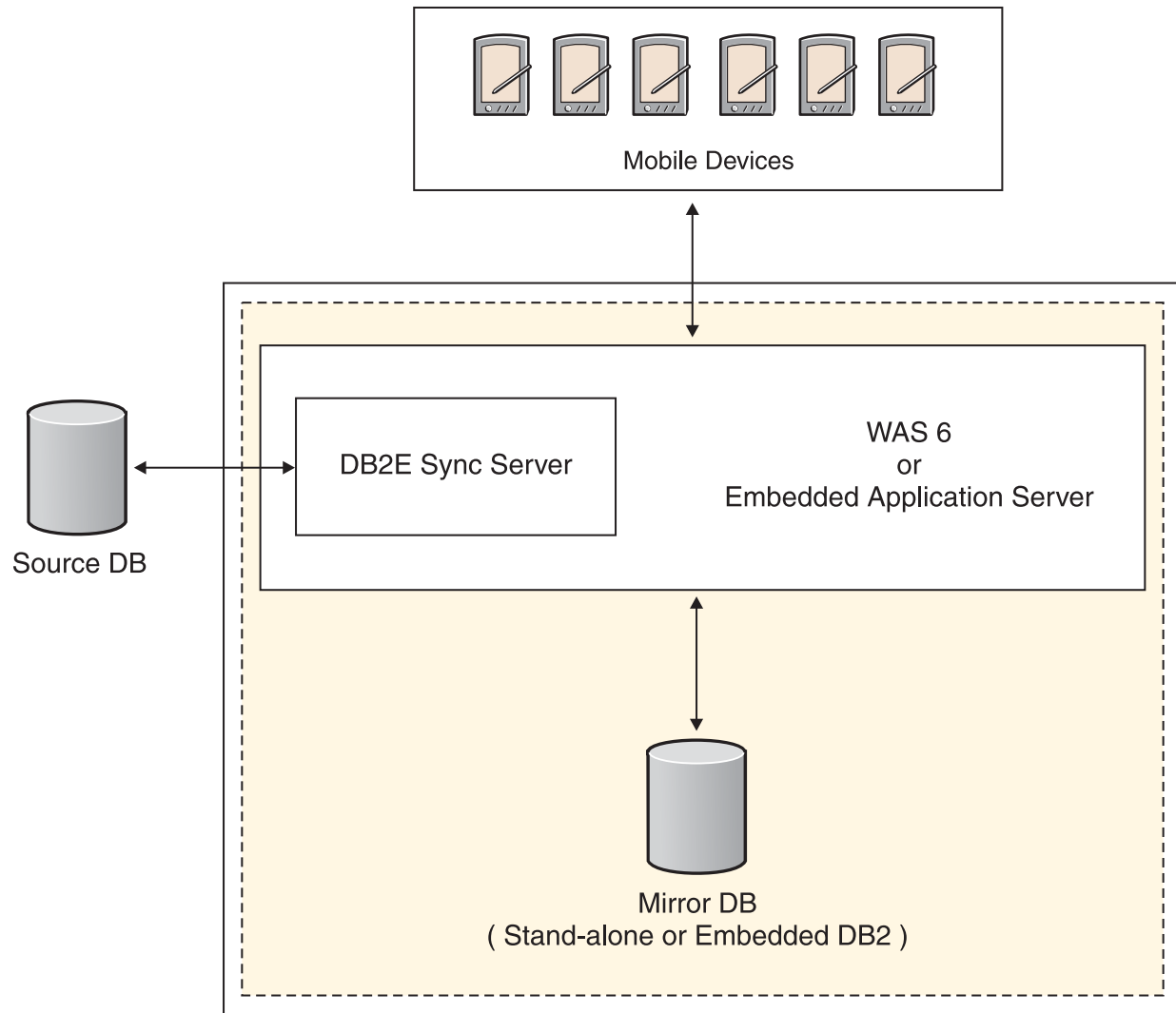
DB2 Everyplace Sync Client

The DB2 Everyplace Sync Client is a component that mobile applications use to synchronize data between the DB2 Everyplace Sync Server. It handles bidirectional synchronization of enterprise relational data with the DB2 Everyplace mobile database. The DB2 Everyplace Sync Client also allows you to easily distribute and update mobile applications on your mobile devices as well as run stored procedures that reside on a DB2 database.

How data is exchanged between DB2 Everyplace and mobile devices

The DB2 Everyplace Sync Server defines relationships and access rights through DB2 Version 9.1 replication objects such as users, groups, subscriptions, and subscription sets. This information as well as a local copy of the source data are kept on the DB2 Everyplace mirror database. The DB2 Everyplace

Sync Server transfers data to mobile devices through the TCP/IP interface that is provided by the application server. Mobile devices can synchronize data through any channel that supports TCP/IP, such as a direct USB connection or an Internet connection.



DB2 Everyplace environments

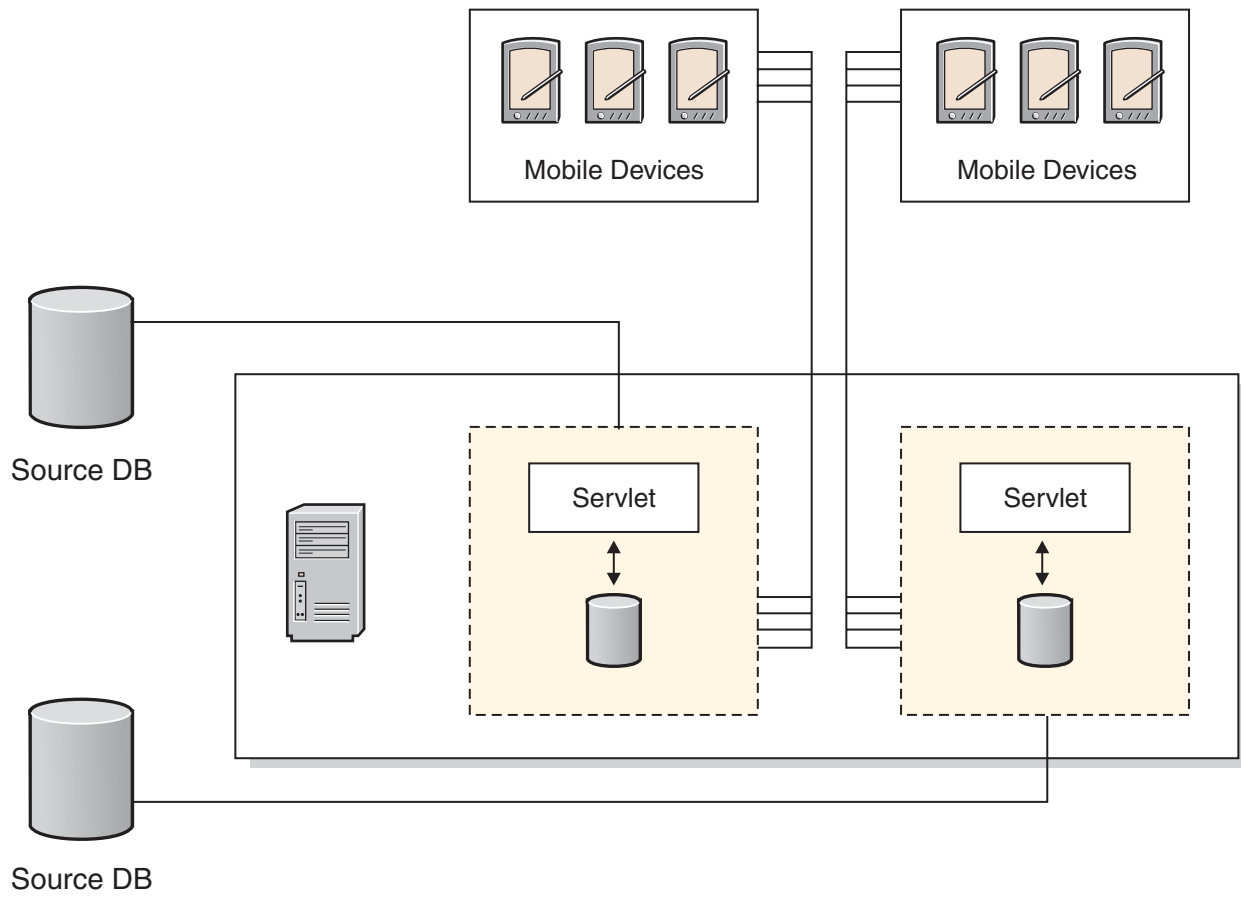
DB2 Everyplace Enterprise Edition is a robust solution for synchronizing enterprise data. You can configure the DB2 Everyplace environment in multiple ways depending on the needs of your network and your users.

Underlying concepts

Like DB2 Version 9.1, DB2 Everyplace operates using instances. A DB2 Everyplace instance consists of the following components:

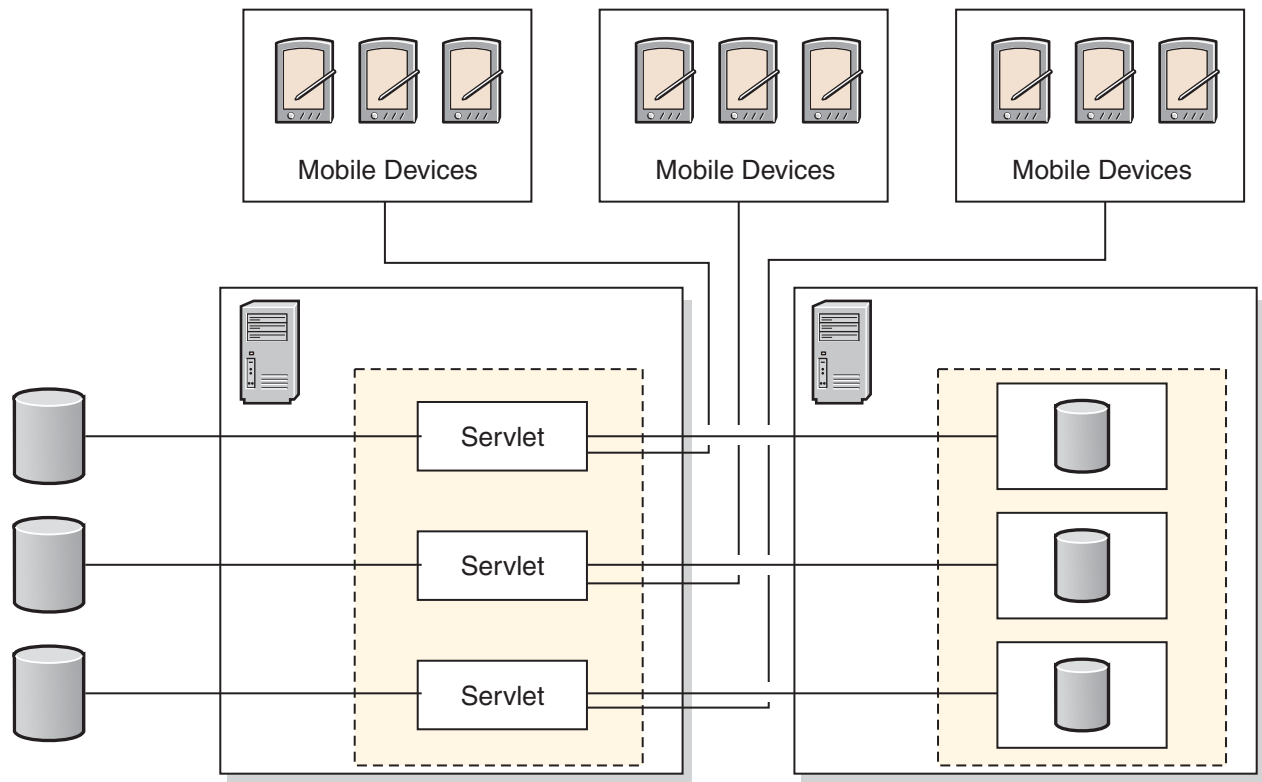
- The DB2 Everyplace Sync Server running in an embedded application server instance
- The DB2 Everyplace mirror database, which is managed by an instance of DB2 Version 9.1

The basic environment



A basic environment places the DB2 Everyplace instance on one physical machine.

The distributed environment



A distributed environment divides the DB2 Everyplace instance into two components:

Distributed server

The distributed server is a machine that runs an instance of the DB2 Everyplace Sync Server inside an embedded application server instance.

Distributed database

The distributed database is a machine that hosts the DB2 Everyplace mirror database on an instance of DB2 Version 9.1.

The clustered environment

A clustered environment allows you to scale the DB2 Everyplace Sync Server across several nodes in a WebSphere Application Server Network Deployment cluster. WAS ND empowers DB2 Everyplace with additional features such as dynamic load balancing, scalability, and fail over.

The remote administration environment

A remote administration environment allows you to administer a DB2 Everyplace Sync Server instance that is located on another physical system by using the Mobile Devices Administration Center. For example, you can install DB2 Everyplace on a Windows workstation, configure it for a remote administration environment, and use it to administer a DB2 Everyplace Sync Server on an AIX system.

The DB2 Everyplace sample applications

The sample applications provide examples of applications that use DB2 Everyplace.

The Visiting Nurse sample application demonstrates bidirectional synchronization between the mobile database and the Sync Server. The sample application has two parts: one part runs on the Sync Server and another part runs on the mobile device that has the DB2 Everyplace mobile database. The sample application on the mobile device demonstrates the database engine functionality in a standalone environment. When the Sync Server sample application and the DB2 Everyplace mobile database engine sample application are used together, they work as a complete application that invokes all components of DB2 Everyplace.

The DB2 Sync sample application demonstrates how to use the IBM Sync Client API to synchronize tables of the subscriptions that are defined in the Mobile Devices Administration Center.

The administration process

If you are just getting started with administration and have not yet created any objects, follow these steps to set up synchronization.

Think about the mobile users within your organization and plan their synchronization needs. What data do they need to access? What applications do they use? The answers to questions like these help you determine how many synchronization groups you need and which users will be associated with each group. See “Creating a data synchronization group” on page 24 for more information about creating groups.

1. Create and assign users to each group in one of two ways:

- Create new users in the Mobile Devices Administration Center and assign these users to groups.
- Import existing user definitions.

If you already have multiple groups set up within your organization, you might be able to import the users assigned to these groups into the Mobile Devices Administration Center and automatically assign them to groups. See “Importing user definitions from another source” on page 24 for more information.

2. Define replication subscriptions for the tables and files that the members of the group need to access. See “Creating a data synchronization group” on page 24 for more information.

3. To allow a group to access the tables and files that they need, create one or more subscription sets that reference the subscriptions that you created. See “Enabling groups to access their data and files” on page 57 for more information.

Managing users and data

This topic presents information about administering DB2 Everyplace clients, servers, objects, applications, and users.

Mobile Devices Administration Center

This topic describes the Mobile Devices Administration Center, a tool you can use to manage access to applications and data.

What is the Mobile Devices Administration Center?

The Mobile Devices Administration Center is a tool that allows you to deliver synchronization services to groups of users who have the same needs for access to applications and data.

The Mobile Devices Administration Center makes it easier to handle multiple users, because you can define data to be replicated for a group of users, yet make some adjustments to the replication definition for individual users as necessary. You can use it to accomplish a series of administration tasks such as setting up groups of users, and configuring data and file replication for groups.

Related concepts

“Mobile Devices Administration Center and synchronization objects” on page 11

This topic describes the different types of synchronization objects and introduces to you the way to work with them.

“The Mobile Devices Administration Center interface” on page 10

This topic explains aspects of the interface that you will encounter in the main window and within object windows.

“Information displayed in the contents pane” on page 16

After you create an object, information about that object appears in the contents pane. Refresh the view to see the most up-to-date information. To display objects in the contents pane, select a folder in the object tree.

Related tasks

“Starting the Mobile Devices Administration Center”

This topic describes how to start the Mobile Devices Administration Center in Windows, Linux and UNIX operating systems.

“Refreshing the view of an object folder” on page 18

When you make changes to an object, those changes might not be immediately displayed in the contents pane. Status or log information might also need refreshing to be up-to-date.

“Filtering the contents pane” on page 17

If you have several synchronization objects in a folder, you can filter the folder’s contents to more easily find what you need. Filtering a folder’s contents displays a subset of the contents according to search criteria that you enter.

“Deleting an object” on page 14

You can delete any object in the Mobile Devices Administration Center. This topic describes how to delete objects.

Starting the Mobile Devices Administration Center

This topic describes how to start the Mobile Devices Administration Center in Windows, Linux and UNIX operating systems.

Prerequisites: For Linux, you must start the DB2 JDBC Applet Server before you start the Mobile Devices Administration Center. Use the `db2jstrt` command.

Start the Mobile Devices Administration Center:

Windows

Click **Start** → **Programs** → **IBM DB2 Everyplace** → **Start Mobile Devices Administration Center**.

Linux

1. Set the DISPLAY environment variable in the shell that you are using.
Example: `export DISPLAY=:0.0`
2. Change the directory to `$DSYINSTDIR/Server/bin`.
3. Start the Mobile Devices Administration Center.
 - On Linux, if the DB2 JDBC Applet Server is running on a port other than 6789, enter the following command:
`dsyadmin.sh <port>`
where `<port>` is the port that the DB2 JDBC Applet Server is using.
 - If DB2 JDBC Applet server is running on the default port, enter the following command:
`dsyadmin.sh`

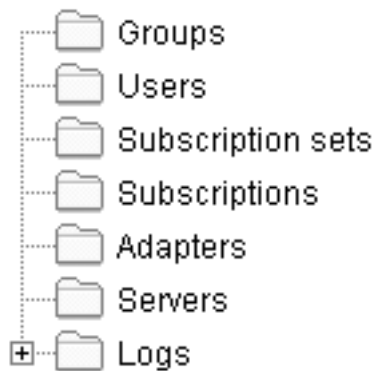
The Mobile Devices Administration Center interface

This topic explains aspects of the interface that you will encounter in the main window and within object windows.

Controls in the main window

The main window contains an object tree and a contents pane to help you manage objects.

Object tree



Use the object tree to display and work with synchronization objects.

Contents pane

Name	Enabled	Description
SAMPLE	No	SAMPLE DE...
DATA1	Yes	DATA1 DES...

Use the contents pane to display and work with individual synchronization objects. The contents pane displays the objects within the object folder that you select in the object tree.

Entry fields

Required fields in are surrounded with a red box. Optional fields are surrounded with a blue box. Additionally, the system warns you with pop-up messages when you are entering invalid information.

Accessing help

To view help files for the Mobile Devices Administration Center, select **Help** → **Synchronization Server Help**.

Windows XP systems

If the Sync Server Help opens in an Internet Explorer browser, the browser displays the following warning:

"To help protect your security, Internet Explorer has restricted this file from showing active content that could access your computer. Click here for options..."

Click the warning message to display the missing content.

Unix systems

To view online help, install the Mozilla Application Suite web browser.

Mobile Devices Administration Center and synchronization objects

This topic describes the different types of synchronization objects and introduces to you the way to work with them.

The data synchronization needs of mobile workers often differ by their job responsibilities. For example, a nursing shift supervisor might need real-time access to information about which beds are taken, as well as status of patients in observation rooms. Doctors working a rotation need access to the list of patients

they are to see that day, as well as status information about the patients. These hospital workers need mobile access to the hospital's enterprise data, but they work with different applications and different sets of data to perform their jobs.

The DB2 Everyplace Sync Server provides an administration tool that helps you manage and deliver synchronization services to groups of users with similar data synchronization needs. With the Mobile Devices Administration Center, you can define what information a group of users is allowed to access and how that information is to be replicated to each user's mobile device.

What is a synchronization object?:

In the Mobile Devices Administration Center, you administer the synchronization process through a set of *synchronization objects*. A synchronization object contains information about aspects of the synchronization process in your organization. You can create and edit groups, subscriptions, subscription sets, and users to handle your organization's synchronization requirements. Logs are available for viewing only.

There are six types of synchronization objects:

Group A group of users with similar mobile data synchronization needs. You define synchronization characteristics for each group, such as which applications the users in the group need to access to perform their jobs and what subsets of enterprise data they need to access.

User A user who uses the DB2 Everyplace Sync Server to synchronize data between a source (the enterprise system) and a target (the mobile device). You assign a user to a group to provide access to the subscriptions defined in the group's subscription sets.

Server

A server is an instance of DB2 Everyplace Sync Server that runs on a host and listens to a port. It synchronizes data between targets and mirror databases. Optionally, you can configure a server for replicating data between mirror and source databases.

Subscription

A specification for what information in a source database or server is to be replicated to a target database (the DB2 Everyplace mobile database on the mobile device). Like a magazine subscription where you choose the types of information you want to see on a periodic basis, a subscription allows you to define which subsets of your enterprise's data and files the group members are allowed to access. Members can then access and synchronize just this subset of data and files, improving both security and performance. You can create two types of subscriptions: file subscriptions for files stored at the source server, and table subscriptions in the source database using either IBM DataPropagator™ or JDBC subscriptions.

Subscription set

A collection of subscriptions. To provide group members with access to the data and files defined in subscriptions, you collect the subscriptions together in a container called a *subscription set*, then assign this container object to the group. This two-step process of enabling members of a group to access the information that they need makes administration easier because you can bundle a set of subscriptions and assign that bundle to multiple groups if necessary.

When users start the synchronization client software on the device, they choose which subscription set to synchronize. The menu of subscription sets that appears on the client is created from the list of subscription sets associated with the user's group.

During a synchronization, if the client failed to synchronize a subscription successfully, it skips the remaining subscriptions in the same subscription set, and continues with the next subscription set. Group closely related subscriptions into a subscription set instead of randomly assigning different subscriptions to a subscription set. This can make troubleshooting easier if a subscription set fails to synchronize successfully.

Adapter

An adapter is used to synchronize and communicate with the DB2 Everyplace Sync Server. A

collection of adapters is included for synchronizing files, relational data with DB2, relational data with JDBC, and remote query and stored procedure functionality.

Log After you implement mobile data synchronization, you can monitor any synchronization problems using the messages written to the log.

Creating synchronization objects:

This topic describes how to create an object from the object tree and when you are working within another object.

Although each object requires different input, you create all objects using the same basic process in the Mobile Devices Administration Center.

Creating an object from the object tree:

This topic describes how to create an object from the object tree.

To create a synchronization object:

Perform the following steps in the Mobile Devices Administration Center.

1. In the object tree, right-click the object folder that corresponds to the type of object that you want to create.
2. Select **Create**. If you are creating a subscription, a submenu displays. Select either **File subscription**, **Table Subscription**, or **Custom subscription**, depending on the type of subscription that you want to create.
3. The Create window opens with the object type displayed in the title of the window.
4. Complete the window tabs for the object. See one of the following topics:
 - For groups, see “Creating a data synchronization group” on page 24.
 - For users, see “Creating users in the Mobile Devices Administration Center” on page 19.
 - For subscriptions sets, see “Enabling groups to access their data and files” on page 57.
 - For JDBC subscriptions, see “Creating a JDBC subscription” on page 33.
 - For upload subscriptions, see “Creating an upload subscription” on page 39.
 - For file subscriptions, see “Creating a file subscription” on page 44.
 - For Custom subscriptions, see “Administering Custom subscriptions” on page 46.
5. **Optional:** After you save an object, you can edit the object to complete it or make changes.

Creating a new object when working with another:

The Mobile Devices Administration Center allows you to create new synchronization objects for a related object without disturbing your workflow.

For example, you might be editing the characteristics of a group named “Visiting Nurses” and realize that you need to add a new nurse who just started work.

To create a new object while you work with another object:

Perform the following steps in the Mobile Devices Administration Center.

1. Leave the Edit Group window open. Click **Create** on the Users tab to open the Create User window.
2. Add the new nurse.
3. Assign the nurse to the Visiting Nurses group.

Important: Refresh the Users folder to see the new nurse in the list of users.

Where synchronization objects are stored:

As you create, edit, or delete synchronization objects, the Mobile Devices Administration Center records this administration information in a control database stored on the source system.

If your server configuration is multiple, then the control database is located on only one server in your server group. The name of this database is DSYCTLDB. DSYCTLDB is a reserved name; you cannot rename the database or use the name DSYCTLDB for any other database in the system.

Created at installation time, the control database helps the DB2 Everyplace Sync Server authenticate users and obtain subscription information to determine what to replicate for the synchronization requests. Because it contains synchronization setup information as well as status information about synchronization operations, ensure that you regularly back up the administration control database. Passwords and other information in this database are not encrypted; thus, ensure that you take appropriate steps to protect this database.

Editing a synchronization object:

After you save a synchronization object in the administration control database, you can open the object in edit mode to add information or make changes. This topic introduces how to edit a synchronization object.

To edit a synchronization object:

Perform the following steps in the Mobile Devices Administration Center.

1. In the object tree, open the object folder that corresponds to the type of object that you want to edit. The contents of the folder are displayed in the contents pane.
2. In the contents pane, right-click the object that you want to edit.
3. Select **Edit**.

The Change window opens with the object type displayed in the title of the window. The Change window contains all the same fields as the Create window for the object. For information about specific tabs and fields, see the topics for the type of object that you want to edit:

- To edit a group, see “Creating a data synchronization group” on page 24.
- To edit a user, see “Creating users in the Mobile Devices Administration Center” on page 19.
- To edit a subscription, see “Creating a data synchronization group” on page 24.
- To edit a subscription set, see “Enabling groups to access their data and files” on page 57.

Important: The administrator should ensure that no synchronization sessions are outstanding when a subscription is edited by blocking the Subscription, and waiting for any in progress synchronization sessions to complete. Failure to do so can result in loss of client changes.

Deleting an object:

You can delete any object in the Mobile Devices Administration Center. This topic describes how to delete objects.

Before you delete an object, however, consider the consequences on your users’ synchronization sessions. Will deleting the object cause a synchronization session to fail? If it does, the objects affected by the deletion are automatically disabled.

When you delete an object, its information is automatically deleted from the administration control database. When you delete a subscription, mirror tables associated with that subscription are automatically dropped from the mirror database as well.

To delete an object:

1. In the object tree of the Mobile Devices Administration Center, open the object folder that corresponds to the type of object that you want to edit.
2. In the contents pane, right-click the object that you want to delete. To select multiple objects serially, hold down the Ctrl key while selecting. To select multiple objects contiguously, hold down the Shift key while selecting.
3. Select **Delete**. A confirmation window displays the objects that you selected for deletion. Clear the **Delete** check box for any items you decide to keep.
4. Click **OK**.

Effects of administrative actions on pending client changes:

Changing or deleting a synchronization object, such as a subscription, group, user, or device, affects changes that are on the mobile device. Resetting or deleting a mobile device also has potential adverse effects.

The effect varies depending on the administrative action. The following table describes the actions and the associated effect on the pending client changes.

Table 1. Administrative actions and their effects

Administrative action	Effects on pending client changes
Edit group: Remove subscription set from group and add subscription set back to group	Client changes for all mobile devices that are assigned to all users in the group are applied to source database.
Edit group: Remove subscription set from group	Client changes for all mobile devices that are assigned to all users in the group are not sent to the server and are not applied to the source database.
Edit group: Remove user from group	Client changes for all mobile devices that are assigned to the user that was removed are not sent to the server and applied to the source database.
Edit group: Remove user from group and add user back to group	Client changes for all mobile devices that are assigned to the user are archived to <DSYPATH>\Server\logs\archive.
Reset group	Client changes for all mobile devices that are assigned to all users in the group are applied to the source database.
Delete group	Client changes for all mobile devices that are assigned to all users in the group are not sent to the server and applied to the source database.
Delete and recreate group	Client changes for all mobile devices for all users in the group are archived to <DSYPATH>\Server\logs\archive.
Edit user: Remove from group	Client changes for all mobile devices that are assigned to the user are not sent to the server and applied to the source database.
Edit user: Remove from group and add back to group	Client changes for all mobile devices that are assigned to the user are applied to the source database.
Reset user	Client changes for all mobile devices that are assigned to the user are applied to the source database.
Delete user	Client changes for all mobile devices that are assigned to the user are not sent to the server and applied to the source database.
Delete and recreate user	Client changes for all mobile devices that are assigned to the user is archived to <DSYPATH>\Server\logs\archive.
Reset device	Client changes for the mobile device are applied to the source database.

Table 1. Administrative actions and their effects (continued)

Administrative action	Effects on pending client changes
Delete device	Client changes for the mobile device are archived to <DSYPATH>\Server\logs\archive.
Edit subscription set: Remove group from subscription set	Client changes for all mobile devices that are associated with the removed group are not sent to the server and are not applied to the source database.
Edit subscription set: Remove group from subscription set and add group back to subscription set	Client changes for all mobile devices that are associated with the group are applied to the source database.
Edit subscription set: Remove subscription from subscription set	Client changes for all mobile devices that are associated with the removed subscription are not sent to the server and applied to the source database.
Edit subscription set: Remove subscription from subscription set and add subscription back to subscription set	Client changes for all mobile devices that are associated with the subscription are applied to the source database.
Delete subscription set	Client changes for all mobile devices that are associated with the subscription set are not sent to the server and are not applied to the source database.
Delete and recreate subscription set	Client changes for all mobile devices that are associated with the subscription set are applied to the source database.
Edit subscription: Remove table	Client changes for all mobile devices that are associated with this subscription and for the removed table are not sent to the server and are not applied to the source database.
Edit subscription: Remove table and add table back	Client changes for all mobile devices that are associated with this subscription and for the table are archived to <DSYPATH>\Server\logs\archive.
Edit subscription > Edit table: Alter local data encryption	Client changes for all mobile devices that are associated with this subscription are applied to the source database.
Edit subscription > Edit table: Alter row filter predicate	Client changes for all mobile devices that are associated with this subscription are applied to the source database.
Edit subscription > Edit table: Alter column constraints	Client changes for all mobile devices that are associated with this subscription and for the altered table are archived to <DSYPATH>\Server\logs\archive.
Edit subscription > Edit table: Enable or disable the filter scope feature	Client changes for all mobile devices that are associated with this subscription are applied to the source database.
Edit subscription > Edit table: Alter table privileges	Client changes for all mobile devices that are associated with this subscription are applied to the source database based on the new privileges.
Edit subscription > Edit table: Alter table schema	Client changes for all mobile devices that are associated with this subscription and for the altered table are archived to <DSYPATH>\Server\logs\archive.
Delete subscription	Client changes for this subscription are not sent to the server and are not applied to the source database.
Delete and recreate subscription	Client changes for all mobile devices that are associated with this subscription are archived to <DSYPATH>\Server\logs\archive.

Information displayed in the contents pane

After you create an object, information about that object appears in the contents pane. Refresh the view to see the most up-to-date information. To display objects in the contents pane, select a folder in the object tree.

Adapter

This field displays the adapter used by the subscription.

Customizer class name

This field displays the Java™ classname of the adapter.

Data filter

This field displays Yes if you have set a data filter for the user or group and No if no data filter has been set.

Description

This field displays the description that you gave the object when you created it, and appears for all objects except logs.

Device type

This field displays the type of device registered to the user. There can be more than one device type depending on how many devices are registered to this user. Do not enter this information using the Mobile Devices Administration Center; the DB2 Everyplace Sync Server obtains the device type when the user registers the device.

Enabled

This field displays Yes if the group or user has been enabled for synchronization and No if it is not enabled.

Encryption Level

This field displays the encryption level used during data transfer.

Group

When the **Users** folder is selected, this field displays the name of the group to which the user is assigned. When the **Subscription sets** folder is selected, this field displays the number of groups associated with the subscription set.

Name

This field displays the unique name that you gave the object when you created it, and appears for all objects except logs.

Signature

This field displays the signature used by the adapter.

Subscriptions

This field displays the number of subscriptions assigned to the subscription set.

Subscription sets

This field displays the number of subscription sets associated with a group or subscription.

Synchronization status

This field displays the synchronization status of the user. The synchronization status is retrieved by selecting a user, right-clicking it, and then selecting **Synchronization status**. The status displayed is real-time and indicates whether a synchronization session is starting, is in progress, or is completed.

Type

This field displays the type of subscription.

Users

This field displays the number of users assigned to a group.

Filtering the contents pane:

If you have several synchronization objects in a folder, you can filter the folder's contents to more easily find what you need. Filtering a folder's contents displays a subset of the contents according to search criteria that you enter.

The Mobile Devices Administration Center displays only those objects meeting the specified filter criteria. After you set the filter for an object, you must explicitly clear the filter to display all elements in an object, or all objects in the tree.

To filter the contents pane:

Perform the following steps in the Mobile Devices Administration Center.

1. Click an object folder.
2. Click the **Filter** button on bottom right toolbar. The Filter window opens, displaying the column headings in the contents pane for the selected object folder. Some columns cannot be filtered.
3. In the **Values** field of the Locate tab, type your search criteria.

Tip: Another way to quickly find what you need is to sort a column in the contents pane alphabetically. To sort a column, click the column heading. The system rearranges the data rows in alphabetical order according to the column that is currently selected.

Refreshing the view of an object folder:

When you make changes to an object, those changes might not be immediately displayed in the contents pane. Status or log information might also need refreshing to be up-to-date.

To refresh the view of an object folder:

1. Right-click the object folder.
2. Click **Refresh** in the pop-up menu.

Administering user accounts

This topic describes how to create and manage users, people who need to synchronize data with an enterprise server. The user uses a mobile device capable of communicating with the DB2 Everyplace Sync Server to replicate data to and from an enterprise system.

What is a user?

This topic introduces the definition of *users* and briefly describes its relationship with *groups*.

A *user* is defined as someone who needs to synchronize data with an enterprise server. The user uses a mobile device capable of communicating with the DB2 Everyplace Sync Server to replicate data to and from an enterprise system. When a user requests synchronization, the user ID and password are authenticated against LDAP repository. The DB2 Everyplace Sync Server determines if the user is defined in the control database and, if so, to what group the user belongs. Data synchronization fails if the user is not assigned to a group.

Users are put into *groups*, which are collections of users who have similar needs for data access and synchronization. Organizing users into groups helps you minimize the effort required to deliver synchronization services to many users in a uniform way.

Evaluating the characteristics of your mobile users

This topic provides a set of criteria that help you categorize users into groups.

Before you begin creating groups with the Mobile Devices Administration Center, think about the types of mobile users in your organization. Users in almost any organization naturally fall into groups according to the type of work that they do. For example, imagine you are a hospital database administrator responsible for delivering data synchronization services to 10 hospitals throughout the country. Visiting nurses in a home health care program and emergency room shift supervisors would naturally fall into two groups because they have different sets of job responsibilities. However, there are other criteria you should consider when structuring your user groups. Ask yourself the following questions to determine if a given set of users should belong to the same group:

- Will these users all be using the same mid-tier system to synchronize enterprise data?

Each mid-tier system has a different installation of the DB2 Everyplace Sync Server. Thus, all members of a group in the Mobile Devices Administration Center must use the same mid-tier system to

synchronize data. For example, in your role as hospital database administrator you might, for performance reasons, have a different DB2 Everyplace Sync Server for each hospital; therefore the users might need to be further divided if their physical locations are different.

- Do these users access the same type of data and files to perform their jobs?

Members of groups must use the same set of replication subscriptions. Thus, users who require access to different files or tables (or even different columns of the same table) must be placed into different groups.

- Do these users have the same access privileges for a given table?

SQL access privileges are defined for each replication source in a subscription, and that subscription is assigned to an entire group. Thus, if users require different types of access to the same replication source, they must be members of different groups.

You might have a group of nurses, for example, who all require access to the same table. However, are some nurses allowed to only insert data, but others are allowed to update or delete data? Are other nurses allowed to only view the data without changing it? For example, you might have a group of nurses in training that you allow to only view the data. When these nurses complete their training, you move them from the Training data synchronization group to another group for which you defined additional SQL privileges.

Creating users in the Mobile Devices Administration Center

This topic introduces how to create users in the Mobile Devices Administration Center.

You can define users to the DB2 Everyplace Sync Server by creating users in the Mobile Devices Administration Center or importing existing definitions directly into the administration control database.

To Create users in the Mobile Devices Administration Center:

Perform the following steps in the Mobile Devices Administration Center.

1. Right-click the **Users** folder and select **Create**. The Create User window opens.
2. Create identities for users.
3. Assign the user to a group.
4. **Optional:** Set passwords for the user.
5. **Optional:** Define horizontal filter parameters at the user level.
6. **Optional:** Define filter parameters for unsubscribed columns at the user level.
7. Click **OK**.

After you create the user, the user must register their device.

Related concepts

“What is a user?” on page 18

This topic introduces the definition of *users* and briefly describes its relationship with *groups*.

Related tasks

“Allowing multiple devices per user” on page 21

You can let one user to synchronize multiple devices with one user ID and password.

“Importing user definitions from another source” on page 24

If you already have user names, passwords, and descriptions defined in another program, you can import this information directly into the administration control database and bypass creating these users in the Mobile Devices Administration Center. Use the XML Scripting Tool to perform this task.

Disabling synchronization

You might choose to disable an object temporarily if you are making administration changes to either the group or the user. This topic introduces how to disable

Disable a group or user in one of the following ways in the Mobile Devices Administration Center.

- By opening the object:
 1. From the object tree, select the object folder. Existing objects are displayed in the contents pane of the Mobile Devices Administration Center.
 2. Right-click the object that you want to disable.
 3. Select **Edit**. The Change window for the object opens.
 4. From any tab of the Change window, clear the **Enable synchronization** check box.
 5. Click **OK**. The **Enabled** field in the contents pane changes from Yes to No.
- From the contents pane (if you want to disable multiple objects at the same time):
 1. From the object tree, select the appropriate object folder.
 2. In the contents pane, right-click the object that you want to disable, then click **Disable**.

Resetting groups, users, and devices

You might have to reset a group, user, or device if it is having a data consistency problem. This might happen if you change a filter on a subscribed table or if a user inserts a row that is not in a filter's scope. Resetting refreshes the data for the device, user, or group during the next synchronization of the client, and does not result in data loss on the client. Resetting a group affects all users and devices in that group. Resetting or deleting a user affects all devices associated with that user. A different user cannot synchronize with the reset device. You must delete a device to let a new user synchronize with it.

Important: Do not reset a group, user, or device while a user is synchronizing.

- To reset a group using the Mobile Device Administration Center:
 1. Click the **Groups** folder on the left-hand pane.
 2. Right-click the group you want to reset on the right-hand pane and select **Reset users**.
 3. Click **OK** in the Confirmation window.
- To reset a user using the Mobile Device Administration Center:
 1. Click the **Users** folder on the left-hand pane.
 2. Right-click the user you want to reset on the right-hand pane and select **Reset users**.
 3. Click **OK** in the Confirmation window.
- To reset a device using the Mobile Device Administration Center:
 1. Click the **Users** folder on the left-hand pane.
 2. Right-click the user you want to reset on the right hand-pane and select **Edit...** The Change User window opens.
 3. On the Devices tab, select the device you want to reset.
 4. Click **Reset**.
 5. Click **OK** in the Confirmation window.
- To reset a group, user, or device using the dsyreset.bat tool from a command line, use the following syntax:

```
DSYReset {[user name]} {-device [device id]} {-group [group name]}
```

where

[user name] is the user name to reset

[device id] is the device id to be reset

[group name] is the group name to reset all users belonging to a group

For example: DSYReset bob would reset the user named 'bob.' DSYReset -group Sales would reset all users belonging to the group 'Sales.'

Deleting groups, users, and devices

You might have to delete a user group, user, or device if it has a technical problem, or if a specific device is being given to another user.

Important: Delete a device only in drastic cases. Data on the device is lost if it does not reach the source database before the delete. If a device is having data inconsistency problems, reset the device. Resetting the device does not result in data loss and is recommended for most cases. If data loss is a concern, run the diagnosis utility to recover data from the message store before deleting a group, user, or device. Do not delete a group, user, or device while a user is synchronizing.

- To delete a group using the Mobile Devices Administration Center:
 1. Click the **Groups** folder on the left-hand pane.
 2. Right-click the group you want to reset on the right-hand pane and select **Delete**.
 3. Click **OK** in the Confirmation window.
- To delete a user using the Mobile Devices Administration Center:
 1. Click on the **Users** folder in the left-hand pane.
 2. Right click the user whose device you want to delete on the right-hand pane and select **Delete**.
 3. Click **OK** in the Confirmation window.
- To delete a device using the Mobile Devices Administration Center:
 1. Click on the **Users** folder in the left-hand pane.
 2. Right click the user whose device you want to reset on the right-hand pane and select **Edit...**
 3. Select a device on the Devices tab of the Change User window.
 4. Click the **Delete** button. If you delete a device, DB2 Everyplace executes the command immediately. If you click **Cancel**, DB2 Everyplace does not return the device.
 5. Click **OK** in the Confirmation window.

Allowing multiple devices per user

You can let one user to synchronize multiple devices with one user ID and password.

A user's device is registered automatically during the first synchronization of the device. The Sync Server allows users to register multiple devices with the same user ID and password depending on the value for the property **AllowMultipleDevicesPerUser** in the control database DSY.PROPERTIES table. The default value is to allow only a single device to register for a user. If the property has been changed to allow multiple devices for a user, you had better not change the setting back to single device per user. The reason is that the behavior after you change the setting back is undefined and unsupported. If you need to do this, delete all the user's devices, and re-register them.

If a user has already been created and assigned to an enabled group, then this user is enabled automatically during the first device's synchronization. If a user wants to use another device, the administrator must take the appropriate action, which is dependant on the value specified in the **AllowMultipleDevicesPerUser** property. You can update the devices in the Mobile Devices Administration Center on the Devices page of the Change User window.

1. AllowMultipleDevicesPerUser=false

If this value is set to false, then you can only have one user per device. In order for a user to synchronize a new device, the administrator must delete the user. When a device is deleted, all information regarding the device is deleted from the control database. If a user tries to synchronize a new device before the administrator deletes the old device from the user, the new device will not be registered.

Note: Synchronize the device before you delete it. Otherwise, you will lose data that has not been synchronized.

2. AllowMultipleDevicesPerUser=true

Allows a user to synchronize an unlimited number of devices without requiring the administrator to delete the device when switching devices. In turn, the control database will keep track of every device this user has synchronized with. By editing a user, the administrator will be able to delete individual devices.

Note: Resetting the user does not remove the user-device association. That is only done by deleting the device.

Related concepts

“What is a user?” on page 18

This topic introduces the definition of *users* and briefly describes its relationship with *groups*.

Related tasks

“Resetting groups, users, and devices” on page 20

You might have to reset a group, user, or device if it is having a data consistency problem. This might happen if you change a filter on a subscribed table or if a user inserts a row that is not in a filter’s scope. Resetting refreshes the data for the device, user, or group during the next synchronization of the client, and does not result in data loss on the client. Resetting a group affects all users and devices in that group. Resetting or deleting a user affects all devices associated with that user. A different user cannot synchronize with the reset device. You must delete a device to let a new user synchronize with it.

“Deleting groups, users, and devices” on page 20

You might have to delete a user group, user, or device if it has a technical problem, or if a specific device is being given to another user.

“Creating users in the Mobile Devices Administration Center” on page 19

This topic introduces how to create users in the Mobile Devices Administration Center.

“Importing user definitions from another source” on page 24

If you already have user names, passwords, and descriptions defined in another program, you can import this information directly into the administration control database and bypass creating these users in the Mobile Devices Administration Center. Use the XML Scripting Tool to perform this task.

Creating mobile users

You need to create a user ID and password for mobile users in order to connect the mobile device to the DB2 Everyplace Sync Server. This topic introduces how you create mobile users in the Mobile Devices Administration Center.

The user ID and password of a mobile user is not a login user ID on the workstation that is running DB2 Everyplace Sync Server. Mobile users can connect to the DB2 Everyplace Sync Server individually or they can be assigned to groups that are associated with applications. This task is part of the greater task of setting up synchronization.

To create a mobile user:

Perform the following steps in the Mobile Devices Administration Center.

1. Right-click the Users folder. Select **Create**.
2. Type a name for the user in the **Name** field. This example uses `testuser`.
3. **Optional:** Type a description for the table subscription in the Description field. This example uses `Test user`.
4. **Optional:** Specify a group for the user in the **Group** field.
5. Click the **Authentication** tab.
6. Enter a password in the **Password** and **Verify password** fields. This example uses `test` as the password.
7. Click **OK** to close the Create User window.

Important: The user is not enabled for synchronization until it is assigned to a group and has connected with the server for the first time.

Related tasks

“Grouping subscriptions into mobile subscription sets” on page 56

Concluding the administration process

This topic provides a brief summary of the administration process.

After you provide administration information, you need to register each user's device with a user ID and password. Then ensure that the user is enabled for synchronization. After users are enabled, they perform an initial synchronization to replicate to the device the configuration information that you defined in the Mobile Devices Administration Center. This information includes what subscription sets the user can synchronize and what data and files are associated with those subscription sets.

Creating identities for users

For authentication purposes, the DB2 Everyplace Sync Server must be able to uniquely identify each user by name. This topic introduces how to create identities for users.

This task is part of the larger task of creating users in the Mobile Devices Administration Center.

Restriction: Do not use the single quotation mark (') in the user name.

To create an identity for a user:

Perform the following steps in the Mobile Devices Administration Center.

1. Click the Identification tab of the Creating upload subscriptions window.
2. Fill in the following fields:

Name: Provide a descriptive, unique name for the user from 1 to 18 characters. The name serves as the user's user ID for accessing the DB2 Everyplace Sync Server on the mid-tier system. Names are case-sensitive.

Description: Provide up to 128 characters of text that describes this subscription.

Assigning a user to a group from the Users window


To access and synchronize data on a source system, the user must belong to a group. If there is no group assignment, the user can register the device, but all other data synchronization fails. A user can belong to only one group at a time. This topic introduces how to assign a user to a group.

If you delete a user from a group or move a user from one group to another, all devices that are associated with that user are deleted.

This task is part of the larger task of creating users in the Mobile Devices Administration Center.

You can also assign users to a group from the Create Group or Edit Group window.

On the Identification tab of the Create User or Edit User window, type the name of the group in the

Group field. To select a group from a list, click the  push button. If no groups are displayed in the list, you must create a group. You can click **Create** to open the Create Group window now, or you can create a group later and associate the group with a user later.

Setting passwords for users

This topic introduces how to set passwords for users.

Important: For increased security, use a long password that includes non-alpha characters. Passwords are case sensitive.

This task is part of the larger task of creating users in the Mobile Devices Administration Center. When you have completed these steps, return to "Creating users in the Mobile Devices Administration Center" on page 19.

To set passwords for a user:

Perform the following steps in the Mobile Devices Administration Center.

1. On the Authentications tab of the Create User window, type the password that a user will use to gain access to the DB2 Everyplace Sync Server.
2. Type the password again in the Verify password field.

Importing user definitions from another source

If you already have user names, passwords, and descriptions defined in another program, you can import this information directly into the administration control database and bypass creating these users in the Mobile Devices Administration Center. Use the XML Scripting Tool to perform this task.

To import user definitions with the XML Scripting Tool:

Use the following command.

```
<AddUser>
  <Name>username</Name>
  <UserGroup>group</UserGroup>
  <Password>12345</Password>
</AddUser>
```

Administering groups of users

This topic introduces the concept of *data synchronization group* (mostly referred as *group*). Also, find in this topic the information about how to create and manage data synchronization groups.

What is a data synchronization group?

A *data synchronization group* is a collection of uniquely named users, or users that have similar data synchronization requirements. Organizing users into groups helps you minimize the effort required to deliver synchronization services to many users in a uniform way.

To allow the users assigned to a group to access the data and files that they need, create replication subscriptions referencing the necessary data and files, collect these subscriptions into a subscription set, and assign the subscription set to the group.

Creating a data synchronization group

This topic introduces how to create a data synchronization group in the Mobile Devices Administration Center.

To create a synchronization group:

Perform the following steps in the Mobile Devices Administration Center.

1. Right-click on the **Groups** folder. Select **Create**. The Create Group window opens.
2. In the Create Group window, type a name for the group in the **Name** field. This example uses Nurse.
3. Optional: Type a description for the group in the Description field. This example uses Nurses Test Group.
4. Click the **Users** tab.
5. Select the user you created from the **Available users** list. This example uses testuser.
6. Click the > push button to move the user to the **Selected users** list.
7. Click the **Subscription Sets** tab.
8. From the **Available subscription sets** list, select a subscription set. The subscription set must be created before you can assign groups to it. This example uses VNurse.
9. Click the > push button to move the subscription set to the **Selected subscription sets** list.
10. Complete the Identification, Clients, and Subscription sets tabs of the Create Group window.

11. Select **Enable Synchronization**.
12. Click **OK** to close the Create Group window.
13. Now you are ready to begin synchronizing data.
14. Assign a user to a group.
15. Click **OK**.

After you create the group, enable it for synchronization so that the group's members can synchronize data.

Related concepts

"What is a data synchronization group?" on page 24

A *data synchronization group* is a collection of uniquely named users, or users that have similar data synchronization requirements. Organizing users into groups helps you minimize the effort required to deliver synchronization services to many users in a uniform way.

"What is a subscription set?" on page 55

"Data filter example" on page 75

"What is a data filter?" on page 62

"What is a user?" on page 18

This topic introduces the definition of *users* and briefly describes its relationship with *groups*.

Related tasks

"Enabling the group to access data and files" on page 26

Bundling subscriptions in a subscription set makes it easier for you to assign a set of subscriptions to multiple groups if necessary. You can assign an unlimited number of subscription sets to a group. To enable the group to access data and files stored on a source system, you assign a subscription set to the group.

"Defining horizontal mirror filter parameters at the group level" on page 69

"Defining filter parameters for unsubscribed columns at the group level" on page 65

"Assigning a user to a group from the Groups window" on page 26

A user can belong to only one group; however, you can move the user from one group to another to accommodate changes in synchronization needs. You can assign an unlimited number of users to a group. This topic introduces how to assign a user to a group from the Groups window.

"Enabling synchronization" on page 26

This topic introduces two approaches you can use to enable synchronization.

Identifying the group

When a user requests synchronization, the DB2 Everyplace Sync Server authenticates the user, and then determines the group to which the user belongs to begin replicating the information referenced in that group's subscriptions. For this reason, the DB2 Everyplace Sync Server must be able to uniquely identify each group by name.

This task is part of the larger task of creating a data synchronization group.

To identify the group:

Perform the following steps in the Mobile Devices Administration Center.

1. Click the **Identification** tab of the Creating upload subscriptions window.
2. Fill in the following fields:

Name: Provide a descriptive, unique name for the group from 1 to 18 characters. Names are case sensitive.

Description: Provide up to 128 characters of text that describes the user.

Enabling the group to access data and files

Bundling subscriptions in a subscription set makes it easier for you to assign a set of subscriptions to multiple groups if necessary. You can assign an unlimited number of subscription sets to a group. To enable the group to access data and files stored on a source system, you assign a subscription set to the group.

To enable the group to access data and files:

Perform the following steps in the Mobile Devices Administration Center.

1. Right-click the **Group** folder and select **Create**. The Create Group window opens.
2. Select the **Subscription sets** tab.
3. In the Available subscription sets list, select the subscription set that you want to assign.

Tip: To select multiple subscription sets serially, hold down the Ctrl key while selecting. To select multiple subscription sets contiguously, hold down the Shift key

If no subscription sets are displayed in the **Available subscription sets** list, you can click **Create** to open the Create Subscription window now, or you can create a subscription set later and associate the group with it later.

4. Click > to assign the selected subscription sets, or click >> to assign all available subscription sets at once. Assigned subscription sets appear in the Selected subscription sets list.

You can also associate groups with a subscription set from within the Subscriptions window.

Assigning a user to a group from the Groups window

A user can belong to only one group; however, you can move the user from one group to another to accommodate changes in synchronization needs. You can assign an unlimited number of users to a group. This topic introduces how to assign a user to a group from the Groups window.

To assign a user to a group from the Groups window:

Perform the following steps in the Mobile Devices Administration Center.

1. Go to the Users tab of the Create Group or Change Group window.
2. In the **Available users** list, select the user that you want to assign to the group. To select multiple users serially, hold down the Ctrl key while selecting; To select multiple users contiguously hold down the Shift key.

If no users are displayed in the **Available users** list, you will need to create a user. You can click **Create** to open the Create User window now, or you can create a user later and assign the user to a group later.

3. Click > to assign the users. Assigned users are displayed in the **Selected users** list. Click >> to assign all available users at once.

Enabling synchronization

This topic introduces two approaches you can use to enable synchronization.

To enable synchronization, choose one of the following methods:

- From the object tree:
 1. Select the object folder. Existing objects are displayed in the contents pane of the Mobile Devices Administration Center.
 2. Right-click the object that you want to enable, and then click **Edit**. The Change window for the object opens.
 3. From any tab of the Change window, select the **Enable synchronization** check box. If the check box appears inactive, you did not provide all the required information for the object to be enabled.
 4. Click **OK**. The **Enabled** field in the contents pane changes from No to Yes.

- From the contents pane (if you want to enable multiple objects at the same time):
 1. In the object tree, open the object folder. Existing objects are displayed in the contents pane of the Mobile Devices Administration Center.
 2. Right-click the object and click **Enable**. If the **Enable** option is not present on the menu, you did not provide all the required information for the object to be enabled.

Administering subscriptions

This topic introduces the concept of *subscription* and the different types of it. Also, find in this topic the information about how to create and manage subscriptions.

Data setup and replication

To enable a group of users to access the data and files that its members require, you must define one or more replication subscriptions. A *replication subscription* provides specifications for how the information in a source system (an enterprise server) is to be synchronized with a target system (the mobile device).

You can create several types of subscriptions in the Mobile Devices Administration Center: JDBC subscriptions, upload subscriptions, file subscriptions, and custom subscriptions.

- *JDBC subscriptions* provide users with access to data in source tables on a data source with a JDBC interface, including DB2, Oracle, Microsoft® SQL Server, and Informix®.
- *Upload subscriptions* only allow the user to directly insert rows into a table on a source database. The source table can reside on any database that supports JDBC. Related tables on the mobile devices are not refreshed during synchronization.
- *File subscriptions* allow replication of any type of file stored at the source server and are not bi-directional. They are from the DB2 Everyplace Sync Server to the mobile device. File subscriptions can be created as a convenient way to distribute and update mobile applications and other data to the devices.
- *Custom subscriptions* allow you to call stored procedures when you synchronize your mobile device with the DB2 Everyplace Sync Server.
- *DataPropagator subscriptions* provide users with access to data in source tables on a DB2 server. You must use the XML scripting tool to create DataPropagator subscriptions.

To determine which subscriptions you need to create, evaluate what information your groups of mobile users currently require to perform their work, what data their applications access and what files they view or use regularly.

The default adapters

DB2 Everyplace Sync Server provides four adapters that you can use to create subscriptions.

- DSYDPROPR (DataPropagator adapter for relational data)
- DSYJDBC (JDBC adapter for relational data)
- DSYFILE (File adapter for copying files)
- Agent Adapter (Remote query and stored procedure adapter for running stored procedures)

Administering JDBC subscriptions

This topic explains the basics of JDBC subscriptions and how to create and configure them in DB2 Everyplace.

What is a JDBC subscription?:

This topic introduces the concept of *JDBC subscriptions*.

A *JDBC subscription* provides specifications for how data in a source table is to be replicated to a table on the target, or mobile, system. For example, the JDBC subscription provides information about access privileges the users of the subscription have to the data and which columns or rows will be replicated during synchronization.

JDBC subscriptions use the trigger-based JDBC adapter for replication.

Specifying source tables, target tables, and privileges for the subscription:

You must define at least one table for every JDBC subscription. If you specified more than one table for the subscription, these tables will be synchronized in the same order that you added them when you created the subscription. The list of tables on the **Define Replication Subscription** window shows this order.

Do not specify the same source and target tables in two different subscriptions. If you want to create two subscriptions that replicate the same source table, use different target tables.

If a source table of a table subscription is to be dropped and recreated, the administrator must first delete that table from the table subscription, and then add that table back to the table subscription after the source table is recreated.

This task is part of the larger task of creating a JDBC subscription.

To add a table to the subscription:

1. On the Identification tab of the Create JDBC Subscription window, click **Define subscription**. The Define Replication Subscription window opens.
2. In the Define Replication Subscription window, click **Add**. The Add Table window opens.
3. In the Add Table window, use the **Table** list to select the replication source to add to this subscription.

Note: Click **Filter** to change the source filter used to limit the available tables in the table list.

The % symbol is a wildcard. Typing % in the Source schema and Source table fields displays all available tables in the table list.

4. In the **Target schema** field, specify the schema that the table will have on the mobile device. If you do not specify a schema, the system uses the schema of the source table by default.
5. In the **Target table** field, specify the name that the table will have on the mobile device. If you do not specify a name, the system uses the name of the source table by default.
6. Under **Access privileges**, select the SQL access privileges that you want users of this subscription to have for this replication source.

For example, if some of your visiting nurses are allowed to insert, update, or delete data, but trainee nurses are allowed to view the data only, you need to create two different subscriptions to handle the difference in access privileges. These subscriptions must use separate mirror tables. For more information about SQL access privileges, see the DB2 Version 9.1 documentation.

7. **Optional:** Select the **Encrypt on device** check box. This enables encryption on the mobile device database.

Note: Encryption is available on Windows®, WinCE,, and Palm OS client platforms only.

8. **Optional:** Select the **Use Constraints defined in selected DB2 source** check box. This enables the constraints from the selected DB2 source table for the target table.
9. **Optional:** Select the **Use Indexes defined in selected DB2 source** check box. This enables the indexes from the selected DB2 source table for the target table.
10. **Optional:** In the **Source table space** field, click the down arrow and select the table space. If you do not select a table space, then DB2 Everyplace will create the table in the default table space.

Note: This option is available only for DB2 and Oracle sources. DB2 on iSeries™ is not supported.

11. **Optional:** In the **Mirror table space** field, click the down arrow and select the table space. If you do not select a table space, then DB2 Everyplace will create the table in the default table space.
12. Click **Add** to add the replication source to the subscription window.
13. Click **Close** to close the Add Table window.

Specifying the mirror database driver and URL:

You must specify the source and mirror databases for a synchronizing subscription. For upload subscriptions, no mirror database is required. This topic introduces how to specify the mirror database.

This task is part of the larger task of creating a JDBC subscription.

To specify the mirror database driver and URL:

Perform the following steps in the Mobile Devices Administration Center.

1. On the Mirror tab of the Create JDBC Subscription window, provide the following information:

Database URL

Select the JDBC Database for the mirror database. Click the  push button to open the Select Mirror Database window.

User ID

Type a valid DB2 Version 9.1 database user ID. Ensure this ID has, at a minimum, the SQL privileges that you granted for the subscriptions defined against the database.

Password

A valid password for the user ID.

Verify Password

Use this field to type the password again exactly as you typed it in the **Password** field.

Mirror Table space

The DB2 Everyplace Sync Server uses this field to create one of its internal control tables for replication. When the source has been registered to DB2 Everyplace Sync Server subscription, you cannot edit this value. Mirror Table Space field is only valid for DB2 and Oracle sources.

Note for OS/390 source: Enter `database_name.tablespace` in a text entry field to specify a database in which to create the table space. As an alternative, you can enter `tablespace` only (no "dot" in between), and DB2 Everyplace will create the table space in the default database.

2. Click **Test connection** to verify the database settings.

Specifying the source database driver and URL:

You must specify the source and mirror databases for a synchronizing subscription. This topic introduces how to specify the source database.

Prerequisite: For upload subscriptions, no mirror database is required.

This task is part of the larger task of creating a JDBC subscription.

To specify the source database driver and URL:

Perform the following steps in the Mobile Devices Administration Center.

1. On the Source tab of the Create JDBC Subscription window, provide the following information:

Database URL

Type the JDBC Database URL. The correct form for the JDBC driver selected in the Driver field is shown. Replace the variables in braces with the information to connect to the database. For example, the default Database URL for DB2 Version 9.1 local installation is `jdbc:db2:{database}`. Replace `{database}` with the name of the database you plan to use.

Driver Select the JDBC database driver to use with this subscription.

User ID

Type a valid source database user ID. Ensure this ID has, at a minimum, the SQL privileges that you granted for the subscriptions defined against the database. For example, if you granted the SQL UPDATE privilege in the subscription, the user ID must have UPDATE privileges defined for the source database.

Notes:

- a. The user requires SYSADM or DBADM authority because subscription creation requires it to execute the SQL: CREATE TABLE, CREATE TRIGGER, CREATE INDEX, and LOCK TABLE.
- b. To create a JDBC subscription to a DB2/400 (on iSeries server) data source, the user ID for database connection requires the *ALLOBJ authority on the iSeries machine.
- c. To create a JDBC subscription with DB2 on OS/390 (on the zSeries server), the user ID for the database connection requires DBADMIN and CREATEIN privileges for the DSY schema.

Password

A valid source database password for the user ID.

Verify Password

Use this field to type the password again exactly as you typed it in the **Password** field.

Control Table space

The DB2 Everyplace Sync Server uses this field to create one of its internal control tables for replication. When the source has been registered to a DB2 Everyplace Sync Server subscription, you cannot edit this value. Control Table Space field is only valid for DB2 and Oracle sources.

Note for OS/390 source: Enter `database_name.tablespace` in a text entry field to specify a database in which to create the table space. As an alternative, you can enter `tablespace` only (no "dot" in between), and DB2 Everyplace will create the table space in the default database.

2. Click **Test connection** to verify the source database settings.

Configuring the DB2 Everyplace Sync Server to access data sources using JDBC:

Before you can create a JDBC or upload subscription in the Mobile Device Administration Center, you must configure the DB2 Everyplace Sync Server to access the data source.

Prerequisites:

Your data source and JDBC driver must be supported by the DB2 Everyplace Sync Server. See the DB2 Everyplace Web site at <http://www.ibm.com/software/data/db2/everyplace> for the list of data sources and JDBC drivers that are supported by the DB2 Everyplace Sync Server.

Important: Remember the following special considerations when you configure the DB2 Everyplace Sync Server to access data sources:

- **DB2 on iSeries:** The DB2 Everyplace Sync Server might time out while it waits for tables and rows of the subscribed tables to become unlocked. This situation can happen when a concurrent source application performs a long transaction. To avoid this situation, set the table and row lock wait times of the subscribed tables to a value that exceeds the execution time of the longest transaction that might be performed by an application.
- **JDBC drivers for DB2 on iSeries:** Be sure to create a collection library using the SQL statement CREATE SCHEMA..

To add the JDBC driver to your system's class path:

1. Exit the Mobile Devices Administration Center if it is running.
2. Stop the DB2 Everyplace Sync Server servlet or the service if it is running.
3. Using a text editor, open \Server\bin\dsysetjavahome.bat file.
4. Add the complete path of the JDBC driver to the SET™ JDBC_DRV_CP line of the file. For example, to configure access to an Oracle source using the JDBC client driver installed to C:\Oracle\jdbc\lib\, add the following path to the SET JDBC_DRV_CP line of the dsysetjavahome.bat file:
SET JDBC_DRV_CP=C:\Oracle\jdbc\lib\classes12.jar

You can specify multiple JDBC client drivers. For example:

```
SET JDBC_DRV_CP=C:\Oracle\jdbc\lib\classes12.jar;E:\Informix\jdbc\lib\ifxjdbc.jar
```

5. Adjust the DB2 Everyplace Sync Server properties by editing \WAS\config\cells\DefaultNode\nodes\DefaultNode\servers\ DB2eServer\Server.xml. For example, add the following line
<classpath>C:\Oracle\jdbc\lib\classes12.jar</classpath>

You can specify multiple JDBC client drivers.

```
<classpath>C:\Oracle\jdbc\lib\classes12.jar</classpath>  
<classpath>E:\Informix\jdbc\lib\ifxjdbc.jar</classpath>
```

6. Restart the DB2 Everyplace Sync Server servlet or the service to make the changes effective.

Specifying the encryption level of a JDBC subscription:

Use this topic to learn the options of the encryption level available.

Encryption support for Windows CE and Windows platforms is only available with the Microsoft High Encryption Pack available from Microsoft. The Microsoft High Encryption Pack must be installed to use encryption between Windows CE/Pocket PC clients or Windows platform clients and the DB2 Everyplace Sync Server. Encryption support on Palm OS requires Palm OS 3.3 or later.

Note: This topic refers to transport encryption only, not encryption on the device and thus should be used when the network is untrusted and you want the data to be private.

This task is part of the larger task of creating a JDBC subscription. When you have completed these steps, return to "Creating a JDBC subscription" on page 33.

To specify the encryption level of a JDBC subscription.

On the Identification page, select the encryption level using the **Encryption level** field. The following three options are available:

None No encryption will be used during data transfer.

Limited: 56-bit Data Encryption Standard

All data transfers will be encrypted using a 56-bit key. This provides basic security with a small performance penalty.

Strong: 128-bit Data Encryption Standard

All data transfers will be encrypted using a 128-bit key. This provides enhanced security with a larger performance penalty than Limited encryption.

Related tasks

“Creating a JDBC subscription” on page 33

Use the **Create JDBC Subscription** window to create a JDBC table subscription. Each table subscription that you create can reference one or more tables.

“Identifying the JDBC subscription”

Use this topic to learn how to identify the JDBC subscription in the Mobile Devices Administration Center.

“Specifying the source database driver and URL” on page 29

You must specify the source and mirror databases for a synchronizing subscription. This topic introduces how to specify the source database.

“Specifying the mirror database driver and URL” on page 29

You must specify the source and mirror databases for a synchronizing subscription. For upload subscriptions, no mirror database is required. This topic introduces how to specify the mirror database.

“Specifying source tables, target tables, and privileges for the subscription” on page 28

“Filtering columns from the source database to the mirror database” on page 63

“Filtering columns from the mirror database to the client device” on page 63

“Filtering rows from the source database to the mirror database” on page 66

“Filtering rows from the mirror database to the client device” on page 67

“Batching replication requests” on page 34

Use this topic to batch replication requests.

“Assigning a subscription set to a JDBC subscription” on page 60

Identifying the JDBC subscription:

Use this topic to learn how to identify the JDBC subscription in the Mobile Devices Administration Center.

After authenticating a user, the DB2 Everyplace Sync Server determines what group the user belongs to, then queries the administration control database to determine what subscription is involved in the request. For this reason, the DB2 Everyplace Sync Server must be able to uniquely identify each subscription by name.

This task is part of the larger task of creating a JDBC subscription. When you have completed these steps, return to “Creating a JDBC subscription” on page 33.

To identify the JDBC subscription:

Perform the following steps in the Mobile Devices Administration Center.

1. Click the **Identification** tab of the Create JDBC Subscription window.
2. Fill in the following fields:

Name: Type a descriptive, unique 1- to 18-character name for the subscription.

Description:Optional: Type up to 128 characters of text that describes this subscription.

Related concepts

“Data setup and replication” on page 27

To enable a group of users to access the data and files that its members require, you must define one or more replication subscriptions. A *replication subscription* provides specifications for how the information in a source system (an enterprise server) is to be synchronized with a target system (the mobile device).

Related tasks

“Identifying the JDBC subscription” on page 32

Use this topic to learn how to identify the JDBC subscription in the Mobile Devices Administration Center.

“Specifying the encryption level of a JDBC subscription” on page 31

Use this topic to learn the options of the encryption level available.

“Specifying the source database driver and URL” on page 29

You must specify the source and mirror databases for a synchronizing subscription. This topic introduces how to specify the source database.

“Specifying the mirror database driver and URL” on page 29

You must specify the source and mirror databases for a synchronizing subscription. For upload subscriptions, no mirror database is required. This topic introduces how to specify the mirror database.

“Specifying source tables, target tables, and privileges for the subscription” on page 28

“Filtering columns from the source database to the mirror database” on page 63

“Filtering columns from the mirror database to the client device” on page 63

“Filtering rows from the source database to the mirror database” on page 66

“Filtering rows from the mirror database to the client device” on page 67

“Batching replication requests” on page 34

Use this topic to batch replication requests.

“Assigning a subscription set to a JDBC subscription” on page 60

Creating a JDBC subscription:

Use the **Create JDBC Subscription** window to create a JDBC table subscription. Each table subscription that you create can reference one or more tables.

To create a JDBC subscription:

Perform the following tasks in the Mobile Devices Administration Center.

1. Right-click on the **Subscriptions** folder and select **Create** → **Table subscription** → **JDBC subscription**. The Create JDBC Subscription window opens.
2. “Identifying the JDBC subscription” on page 32
3. “Specifying the encryption level of a JDBC subscription” on page 31
4. “Specifying the source database driver and URL” on page 29
5. “Specifying the mirror database driver and URL” on page 29
6. “Specifying source tables, target tables, and privileges for the subscription” on page 28
7. Optional: “Filtering rows from the source database to the mirror database” on page 66
8. Optional: “Filtering rows from the mirror database to the client device at the subscription level” on page 68
9. Optional: “Filtering columns from the source database to the mirror database” on page 63
10. Optional: “Filtering columns from the mirror database to the client device at the subscription level” on page 64
11. “Assigning a subscription set to a JDBC subscription” on page 60
12. Click **OK**.

Related concepts

“Data setup and replication” on page 27

To enable a group of users to access the data and files that its members require, you must define one

or more replication subscriptions. A *replication subscription* provides specifications for how the information in a source system (an enterprise server) is to be synchronized with a target system (the mobile device).

Batching replication requests:

Use this topic to batch replication requests.

This task is part of the larger task of creating a JDBC subscription. When you have completed these steps, return to “Creating a JDBC subscription” on page 33.

All subscriptions defined against the same mirror database share the same values for batch parameters. Any changes that you make affect all subscriptions defined against the mirror database.

To batch replication requests:

Perform the following steps in the Mobile Devices Administration Center.

1. Ensure a source and mirror database appear on the **Identification** tab of the **Create JDBC subscription** window.
2. Click **Define Subscription**. The Define Replication Subscription window opens.
3. Click **Timing**. The Subscription Timing window opens.
4. In the **Batch window** field, specify a number of seconds that you want the DB2 Everyplace Sync Server to wait between replication cycles. You can also specify 0, and initiate replication manually, via `dsyreplicate.bat`.
5. Click **OK**.

Related concepts

“Data setup and replication” on page 27

To enable a group of users to access the data and files that its members require, you must define one or more replication subscriptions. A *replication subscription* provides specifications for how the information in a source system (an enterprise server) is to be synchronized with a target system (the mobile device).

Related tasks

“Creating a JDBC subscription” on page 33

Use the **Create JDBC Subscription** window to create a JDBC table subscription. Each table subscription that you create can reference one or more tables.

“Identifying the JDBC subscription” on page 32

Use this topic to learn how to identify the JDBC subscription in the Mobile Devices Administration Center.

“Specifying the encryption level of a JDBC subscription” on page 31

Use this topic to learn the options of the encryption level available.

“Specifying the source database driver and URL” on page 29

You must specify the source and mirror databases for a synchronizing subscription. This topic introduces how to specify the source database.

“Specifying the mirror database driver and URL” on page 29

You must specify the source and mirror databases for a synchronizing subscription. For upload subscriptions, no mirror database is required. This topic introduces how to specify the mirror database.

“Specifying source tables, target tables, and privileges for the subscription” on page 28

“Filtering columns from the source database to the mirror database” on page 63

“Filtering columns from the mirror database to the client device” on page 63

“Filtering rows from the source database to the mirror database” on page 66

“Filtering rows from the mirror database to the client device” on page 67

“Assigning a subscription set to a JDBC subscription” on page 60

Restrictions for table subscriptions:

DB2 Everyplace has several restrictions on table subscriptions. Consider the following limitations when you plan and deploy your mobile applications.

- If you are using a source database that is running on Informix Dynamic Server, each table in a subscription should be created by using an explicit schema qualifier.
- If you are adding a table to a subscription, its source and target schema names, column names, and table names cannot be keywords, reserved words, or special registers in SQL, DB2 Version 9.1 and the source database that you are using.
- If the structure of a source table needs to be changed with either an ALTER TABLE statement or DROP TABLE and CREATE TABLE statements, follow these steps:
 1. Remove the table from all subscriptions.
 2. Execute the ALTER TABLE statement or the DROP TABLE and CREATE TABLE statements.
 3. Add the table back to the subscriptions.
- The DB2 Everyplace Sync Client does not support encryption for more than one target database.
- The IBM Cloudscape™ client does not support all features that the DB2 Everyplace client supports. For example, multiple server, ordering of subscription sets/subscriptions/tables, and local data encryption are not supported.
- Referential integrity is not supported for DataPropagator table subscriptions.
- Always replicate DataPropagator table subscriptions on the mirror database server. This means that if replication needs to be performed during administrative actions, the Mobile Devices Administration Center must be run on the mirror database server.
- A DataPropagator subscription cannot use the same mirror database that is used by a JDBC subscription.
- DB2 Everyplace does not support database object names that must be enclosed in double quotes.
- In general, the maximum size of a row in a table is limited by the data source. Adding the table to a JDBC or DataPropagator subscription further restricts the maximum row size. The additional restriction on the maximum row size is approximately 125 bytes.
- The DB2 Everyplace Sync Server does not synchronize times or timestamps with an hour of 24 correctly due to differences in how time is represented in Java and a data source. A time of "24:00:00" is converted to "00:00:00" and causes the data saved in the mobile database to differ from the data saved in the source database. You should avoid using such an hour in your applications.
- A mirror database can replicate with only one source database. DB2 Everyplace does not allow a mirror database to be associated with multiple source databases.
- DB2 Everyplace converts table and column names from the source database to uppercase when applying them in the mirror and target databases. In rare cases, this can result in an inability to create a table due to non-uniqueness of the table name or column name. In other rare cases it can result in a conversion of a character to multiple characters, as in the German “sharp s” (ß), which will be converted to “SS”.
- When a JDBC subscription is created on DB2 Version 9.1 for z/OS, DB2 Everyplace creates tables and indices on these tables in the source database. DB2 Everyplace creates the indices with the default attributes. Thus, DB2 Version 9.1 for z/OS assigns the index data sets to the default storage groups and gives them default space attributes. Because this might affect the performance of mass inserts, you can improve performance by altering the attributes of the indices that start with the prefix DSY.

Example: To alter the attributes of the DSYI19099874167 index to the following parameters:

- Maximum addressability of each data set: 256 megabytes
- Storage group for the data set: DB2ERSTG
- Minimum primary space allocation for the data set: 80 000 kilobytes
- Minimum secondary space allocation data set: 40 000 kilobytes

issue the following command:

```
ALTER INDEX DSYI19099874167 PIECESIZE 256M using
STOGROUP DB2ERSTG
PRIQTY 80000
SECQTY 40000
```

Referential integrity constraints

If your source tables have referential integrity constraints, observe the following restrictions to avoid synchronization and replication failures due to referential integrity constraint violations:

- Do not update the primary keys on the client database or mobile device.
- Do not subscribe tables which have parent-child relationships that involve cycles (for example self-loops).
- When you create subscriptions, add tables in parent-to-child order.
- Referential integrity relationships cannot cross any subscription boundaries.
- Do not subscribe to tables that have triggers, unless the tables are the target of an upload subscription

Administering upload subscriptions

This topic explains the basics of upload subscriptions and how to create and configure them in DB2 Everyplace.

What is an upload subscription?: An upload subscription is a type of JDBC subscription that allows you to insert rows directly into a table on a source database. An upload subscription defines a one-way synchronization from mobile devices to a source database. Normally, no data is sent from the source database to a mobile device during synchronization and refresh. The only exception is that reject notifications are sent to mobile device in the case where some rows failed to be inserted into the source database. In this case, the mobile device is notified that the problematic rows have been rejected by the source database.

The source table can reside on any database that supports JDBC. Upload subscriptions can improve DB2 Everyplace Sync Server's throughput, simplify administration, and increase system reliability.

Upload subscriptions can be used in scenarios where the client devices are used only for collecting data only. Such scenarios include:

- Registering gas or water meter readings.
- Issuing traffic tickets.
- Collecting bar codes in supermarkets.

There are few restrictions on the source database except that it must support JDBC and SQL INSERT statements. Because there is no replication for an upload subscription, the burden on the DB2 Everyplace Sync Server is alleviated.

In an upload subscription, when you synchronize with the DB2 Everyplace Sync Server, the DB2 Everyplace database engine on the handheld device communicates directly with the source database to insert the rows that have been added to the DB2 Everyplace mobile database. The flow of the data is one-way; the tables on the device are not refreshed. No mirror database is involved in this process.

If a primary key is used, multiple inserts that cause a primary conflict are rejected. The rejections are logged in both the client and the Mobile Devices Administration Center on the DB2 Everyplace Sync Server. Other operations, like DELETE and UPDATE, are either ignored by the source database or rejected. Table 2 on page 37 lists client operations and results.

Table 2. Client's possible operations and their consequences

Operation	Results
Client's INSERT operations without conflicts in the primary key	Inserted into the source database
Client's INSERT operations with conflicts in the primary key	Rejected with error messages logged in the client and the Mobile Devices Administration Center
Client's UPDATE operations	Rejected with error messages logged in the client and the Mobile Devices Administration Center
Client's DELETE operations	Ignored silently
First-time synchronization	Tables created in the client with no rows

Use the Create Upload Subscription window to create an upload table subscription. See "Creating an object from the object tree" on page 13 for information about how to open the window. Each table subscription that you create can reference one or more tables.

Related tasks

"Creating an upload subscription" on page 39

"Identifying the upload subscription" on page 40

"Specifying the encryption level of an upload subscription" on page 39

"Specifying the source database driver and URL"

"Specifying source tables and target tables" on page 38

"Assigning a subscription set to an upload subscription" on page 60

Specifying the source database driver and URL:

This task is part of the larger task of creating an upload subscription. When you have completed these steps, return to "Creating an upload subscription" on page 39.

You must specify only the source database for an upload subscription because no mirror database is involved.

To specify the source database driver and URL:

Perform the following steps in the Mobile Devices Administration Center.

On the Source tab of the Create Upload Subscription window, provide the following information:

Database URL

Type the JDBC Database URL. The correct form for the JDBC driver selected in the Driver field is shown. Replace the variables in braces with the information to connect to the database. For example, the default Database URL for DB2 Version 9.1 local installation is `jdbc:db2:database`. Replace *database* with the name of the database you plan to use.

Driver Use this field to select the JDBC database driver to use with this subscription.

User ID

Type a valid source database user ID. Ensure this ID has the SQL INSERT privilege against the source database.

Password

A valid source database password for the user ID.

Verify Password

Type again the password exactly as you typed it in the **Password** field.

Related concepts

“What is an upload subscription?” on page 36

Related tasks

“Creating an upload subscription” on page 39

“Identifying the upload subscription” on page 40

“Specifying the encryption level of an upload subscription” on page 39

“Specifying source tables and target tables”

“Assigning a subscription set to an upload subscription” on page 60

Specifying source tables and target tables:

Restrictions:

- You must define at least one table for every upload subscription.
- The DB2 Everyplace Sync Server does not synchronize times or timestamps with an hour of 24 correctly due to differences in how time is represented in Java and a data source. A time of "24:00:00", for example, is converted to "00:00:00" and causes the data saved in the mobile database to differ from the data saved in the source database. Avoid using such an hour in your applications.
- If you are adding a table to a subscription, its source and target schema names, column names, and table names cannot be keywords, reserved words, or special registers in SQL, DB2, or the source database that you are using.

This task is part of the larger task of creating an upload subscription. When you have completed these steps, return to “Creating an upload subscription” on page 39.

To add a table to the subscription:

1. On the **Identification** page of the Create Upload Subscription, click **Define subscription**. The Define Upload Subscription window opens.
2. In the Define Upload Subscription window, click **Add**. The Add Table window opens.
3. In the Add Table window, use the **Table** list to select the upload source to add to this subscription.

Note: Click **Filter ...** to change the source filter used to limit the available tables in the table list.

The % symbol is a wildcard. Typing % in the Source schema and Source table fields displays all available tables in the table list.

4. In the **Target schema** field, specify the schema that the table will have on the mobile device. If you do not specify a schema, the system uses the schema of the source table, converted to upper case, by default. In rare cases, this can result in an inability to create a table due to non-uniqueness of the table name or column name. In other rare cases it can result in a conversion of a character to multiple characters, as in the German “sharp s” (ß), which will be converted to “SS”.
5. In the **Target table** field, specify the name that the table will have on the mobile device. If you do not specify a name, by default the system uses the name of the source table.
6. Optional: Select the **Encrypt on device** checkbox. This enables encryption on the mobile device database.
7. Optional: Select the **Use Constraints defined in selected DB2 source** checkbox. This enables the constraints from the selected DB2 source table for the target table.
8. Optional: Select the **Use Indexes defined in selected DB2 source** checkbox. This enables the indexes from the selected DB2 source table for the target table.
9. Click **Add** to add the upload source to the subscription. Click **Close** to close the Add Table window.

Related concepts

“What is an upload subscription?” on page 36

Related tasks

“Creating an upload subscription” on page 39

- “Identifying the upload subscription” on page 40
- “Specifying the encryption level of an upload subscription”
- “Specifying the source database driver and URL” on page 37
- “Assigning a subscription set to an upload subscription” on page 60

Specifying the encryption level of an upload subscription:

Encryption support for Windows CE and Windows platforms is only available with the Microsoft High Encryption Pack available from Microsoft. The Microsoft High Encryption Pack must be installed to use encryption between Windows CE/Pocket PC clients or Windows platform clients and the DB2 Everyplace Sync Server. Encryption support on Palm OS requires Palm OS 3.3 or later.

This task is part of the larger task of creating an upload subscription. When you have completed these steps, return to “Creating an upload subscription.”

On the Identification page, select the encryption level from the **Encryption level** drop-down list. The following three options are available:

None No encryption will be used during data transfer.

Limited: 56-bit Data Encryption Standard

All data transfers will be encrypted using a 56-bit key. This provides basic security with a small performance penalty.

Strong: 128-bit Data Encryption Standard

All data transfers will be encrypted using a 128-bit key. This provides enhanced security with a larger performance penalty than Limited encryption.

Related concepts

“What is an upload subscription?” on page 36

Related tasks

- “Creating an upload subscription”
- “Identifying the upload subscription” on page 40
- “Specifying the source database driver and URL” on page 37
- “Specifying source tables and target tables” on page 38
- “Assigning a subscription set to an upload subscription” on page 60

Creating an upload subscription:

To create an upload subscription:

Perform the following steps in the Mobile Devices Administration Center.

1. Right-click on the **Subscriptions** folder and select **Create** → **Table subscription** → **upload subscription**. The Create Upload Subscription window opens.
2. “Identifying the upload subscription” on page 40
3. “Specifying the encryption level of an upload subscription”
4. “Specifying the source database driver and URL” on page 37
5. “Specifying source tables and target tables” on page 38
6. “Assigning a subscription set to an upload subscription” on page 60
7. Click **OK**.

Related concepts

“What is an upload subscription?” on page 36

Related tasks

- “Identifying the upload subscription”
- “Specifying the encryption level of an upload subscription” on page 39
- “Specifying the source database driver and URL” on page 37
- “Specifying source tables and target tables” on page 38
- “Assigning a subscription set to an upload subscription” on page 60

Identifying the upload subscription:

After authenticating a user, the DB2 Everyplace Sync Server determines what group the user belongs to, then queries the administration control database to determine what subscription is involved in the request. For this reason, the DB2 Everyplace Sync Server must be able to uniquely identify each subscription by name.

This task is part of the larger task of creating an upload subscription. When you have completed these steps, return to “Creating an upload subscription” on page 39.

To identify an upload subscription:

Perform the following steps in the Mobile Devices Administration Center.

1. Click the **Identification** tab of the Create Upload Subscription window.
2. Fill in the following fields:
 - Name:** Type a descriptive, unique 1 to 18 character name for the subscription.
 - Description:** Type up to 128 characters of text that describes this subscription.

Related concepts

“What is an upload subscription?” on page 36

Related tasks

- “Creating an upload subscription” on page 39
- “Specifying the encryption level of an upload subscription” on page 39
- “Specifying the source database driver and URL” on page 37
- “Specifying source tables and target tables” on page 38
- “Assigning a subscription set to an upload subscription” on page 60

Administering File subscriptions

This topic explains the basics of file subscriptions and how to create and configure them in DB2 Everyplace.

What is a file subscription?:

This topic introduces the concept of *file subscriptions*.

To provide users access to the files that they require, you define one or more *file subscriptions*. Unlike a DataPropagator subscription, which can contain more than one table, a file subscription can contain only a single file. Thus, if your group of users needs access to several files, you must create several file subscriptions then assign them to an application. A file cannot be referenced in multiple subscriptions.

The file to which you subscribe the group can be any file stored at the source server. For example, it could be a resource file required for an application to function, or it could be a flat file containing the minutes from a morning meeting.

Related tasks

- “Identifying the file subscription” on page 43
- After authenticating a user, the DB2 Everyplace Sync Server determines the group that the user

belongs to, then queries the administration control database to determine what subscription is involved in the request. For this reason, the DB2 Everyplace Sync Server must be able to uniquely identify each subscription by name.

“Specifying the encryption level of a file subscription” on page 44

Use this topic to learn the options of the encryption level available.

“Specifying the name of the source file” on page 42

Use this topic to specify the name of the source file.

“Specifying subscription sets that use the subscription”

After you define a subscription, you must specify the subscription sets that make use of it. When you associate a subscription with a subscription set, you can then assign the subscription set to a group to provide its members with access to the data and files defined in the subscriptions that the subscription set references.

“Flagging a subscription that references a changed file” on page 42

When you make changes to a file referenced in a subscription, you need to flag that the source file has changed so that the DB2 Everyplace Sync Server replicates the file to the subscribed users on the next synchronization.

Specifying subscription sets that use the subscription:

After you define a subscription, you must specify the subscription sets that make use of it. When you associate a subscription with a subscription set, you can then assign the subscription set to a group to provide its members with access to the data and files defined in the subscriptions that the subscription set references.

This task is part of the larger task of creating a file subscription. When you have completed these steps, return to “Creating a file subscription” on page 44.

You can also associate subscriptions with a subscription set using the Subscription set window. For more information, see “Specifying what information the subscription set accesses” on page 58.

To specify subscription sets:

Perform the following steps in the Mobile Devices Administration Center.

- Go to the Subscription sets tab of the Create File Subscription window. In the Available subscription sets list, select the subscription set that you want to associate the subscription with. To select multiple subscription sets serially, hold down the Ctrl key while selecting; To select multiple subscription sets contiguously hold down the Shift key.
- If no subscription sets are displayed in the **Available subscription sets** list, create a subscription set. You can click **Create** to open the Create Subscription set window now, or you can create a subscription set later and associate it with a subscription later. See “Enabling groups to access their data and files” on page 57 for more information about creating a subscription set.
- Click > to assign the subscription sets, or click >> to assign all available subscription sets at once. Assigned subscription sets are displayed in the Selected subscription sets list.

Related tasks

“Creating a file subscription” on page 44

File subscriptions provide users access to the files that they require. Unlike a DataPropagator subscription, which can contain more than one table, a file subscription can contain only one file.

“Identifying the file subscription” on page 43

After authenticating a user, the DB2 Everyplace Sync Server determines the group that the user belongs to, then queries the administration control database to determine what subscription is involved in the request. For this reason, the DB2 Everyplace Sync Server must be able to uniquely identify each subscription by name.

“Specifying the encryption level of a file subscription” on page 44
Use this topic to learn the options of the encryption level available.

“Specifying the name of the source file”
Use this topic to specify the name of the source file.

“Flagging a subscription that references a changed file”
When you make changes to a file referenced in a subscription, you need to flag that the source file has changed so that the DB2 Everyplace Sync Server replicates the file to the subscribed users on the next synchronization.

Flagging a subscription that references a changed file:

When you make changes to a file referenced in a subscription, you need to flag that the source file has changed so that the DB2 Everyplace Sync Server replicates the file to the subscribed users on the next synchronization.

It is a convenient way to update or upgrade applications and data files on the mobile devices.

1. Open the Subscriptions object folder from the object tree in the Mobile Devices Administration Center.
2. In the contents pane, right-click the file subscription and select Reset.

Related tasks

“Creating a file subscription” on page 44
File subscriptions provide users access to the files that they require. Unlike a DataPropagator subscription, which can contain more than one table, a file subscription can contain only one file.

“Identifying the file subscription” on page 43
After authenticating a user, the DB2 Everyplace Sync Server determines the group that the user belongs to, then queries the administration control database to determine what subscription is involved in the request. For this reason, the DB2 Everyplace Sync Server must be able to uniquely identify each subscription by name.

“Specifying the encryption level of a file subscription” on page 44
Use this topic to learn the options of the encryption level available.

“Specifying the name of the source file”
Use this topic to specify the name of the source file.

“Specifying subscription sets that use the subscription” on page 41
After you define a subscription, you must specify the subscription sets that make use of it. When you associate a subscription with a subscription set, you can then assign the subscription set to a group to provide its members with access to the data and files defined in the subscriptions that the subscription set references.

Specifying the name of the source file:

Use this topic to specify the name of the source file.

This task is part of the larger task of creating a file subscription. When you have completed these steps, return to “Creating a file subscription” on page 44.

A *source file* is a file residing on a source system that is synchronized to a target device. A *target file* is a file on a target system receiving the contents of a source file during synchronization.

The source file must be available for selection from the mid-tier system where you invoke the Mobile Devices Administration Center. Thus, before you define the subscription, either connect to the source system where the file is stored or copy the file from the source system to the mid-tier system. If your synchronization configuration has only two tiers, the file is probably already on the system where you invoke the Mobile Devices Administration Center.

To specify the name of the source file:

Perform the following steps in the Mobile Devices Administration Center.

1. Specify the name of the source file in the **Source filename** field on the Identification tab of the Create File Subscription window.
2. If you update the file, you must flag the source file to be replicated again to each of the subscribed users. See “Flagging a subscription that references a changed file” on page 42 for more information.
3. Because Palm OS mobile devices store the file name within the file itself, the **Target filename** field is ignored.

Related tasks

“Creating a file subscription” on page 44

File subscriptions provide users access to the files that they require. Unlike a DataPropagator subscription, which can contain more than one table, a file subscription can contain only one file.

“Identifying the file subscription”

After authenticating a user, the DB2 Everyplace Sync Server determines the group that the user belongs to, then queries the administration control database to determine what subscription is involved in the request. For this reason, the DB2 Everyplace Sync Server must be able to uniquely identify each subscription by name.

“Specifying the encryption level of a file subscription” on page 44

Use this topic to learn the options of the encryption level available.

“Specifying subscription sets that use the subscription” on page 41

After you define a subscription, you must specify the subscription sets that make use of it. When you associate a subscription with a subscription set, you can then assign the subscription set to a group to provide its members with access to the data and files defined in the subscriptions that the subscription set references.

“Flagging a subscription that references a changed file” on page 42

When you make changes to a file referenced in a subscription, you need to flag that the source file has changed so that the DB2 Everyplace Sync Server replicates the file to the subscribed users on the next synchronization.

Identifying the file subscription:

After authenticating a user, the DB2 Everyplace Sync Server determines the group that the user belongs to, then queries the administration control database to determine what subscription is involved in the request. For this reason, the DB2 Everyplace Sync Server must be able to uniquely identify each subscription by name.

This task is part of the larger task of creating a file subscription. When you have completed these steps, return to “Creating a file subscription” on page 44.

To identify a file subscription:

Perform the following steps in the Mobile Devices Administration Center.

1. Click the **Identification** tab of the Creating upload subscriptions window.
2. Fill in the following fields:

Name: Provide a descriptive, unique name for the subscription from 1 to 18 characters long. Names are case-sensitive.

Description: Type up to 128 characters of text that describes this subscription.

Related tasks

“Creating a file subscription” on page 44

File subscriptions provide users access to the files that they require. Unlike a DataPropagator subscription, which can contain more than one table, a file subscription can contain only one file.

“Specifying the encryption level of a file subscription” on page 44

Use this topic to learn the options of the encryption level available.

“Specifying the name of the source file” on page 42
Use this topic to specify the name of the source file.

“Specifying subscription sets that use the subscription” on page 41
After you define a subscription, you must specify the subscription sets that make use of it. When you associate a subscription with a subscription set, you can then assign the subscription set to a group to provide its members with access to the data and files defined in the subscriptions that the subscription set references.

“Flagging a subscription that references a changed file” on page 42
When you make changes to a file referenced in a subscription, you need to flag that the source file has changed so that the DB2 Everyplace Sync Server replicates the file to the subscribed users on the next synchronization.

Specifying the encryption level of a file subscription:

Use this topic to learn the options of the encryption level available.

This task is part of the larger task of creating a file subscription. When you have completed these steps, return to “Creating a file subscription.”

Encryption support for Windows CE and Windows platforms is only available with the Microsoft High Encryption Pack available from Microsoft. The Microsoft High Encryption Pack must be installed to use encryption between Windows CE/Pocket PC clients or Windows platform clients and the DB2 Everyplace Sync Server. Encryption support on Palm OS requires Palm OS 3.3 or later.

On the Identification page, select the encryption level using the **Encryption level** field. The following three options are available:

- **None:** No encryption will be used during data transfer.
- **Limited 56-bit Data Encryption Standard:** All data transfers will be encrypted using a 56-bit key. This provides basic security with a small performance penalty.
- **Strong: 128-bit Data Encryption Standard:** All data transfers will be encrypted using a 128-bit key. This provides enhanced security with a larger performance penalty than Limited encryption.

Related tasks

“Creating a file subscription”

File subscriptions provide users access to the files that they require. Unlike a DataPropagator subscription, which can contain more than one table, a file subscription can contain only one file.

“Identifying the file subscription” on page 43

After authenticating a user, the DB2 Everyplace Sync Server determines the group that the user belongs to, then queries the administration control database to determine what subscription is involved in the request. For this reason, the DB2 Everyplace Sync Server must be able to uniquely identify each subscription by name.

“Specifying the name of the source file” on page 42
Use this topic to specify the name of the source file.

“Specifying subscription sets that use the subscription” on page 41
After you define a subscription, you must specify the subscription sets that make use of it. When you associate a subscription with a subscription set, you can then assign the subscription set to a group to provide its members with access to the data and files defined in the subscriptions that the subscription set references.

“Flagging a subscription that references a changed file” on page 42
When you make changes to a file referenced in a subscription, you need to flag that the source file has changed so that the DB2 Everyplace Sync Server replicates the file to the subscribed users on the next synchronization.

Creating a file subscription:

File subscriptions provide users access to the files that they require. Unlike a DataPropagator subscription, which can contain more than one table, a file subscription can contain only one file.

The file to which you subscribe the group can be any file stored at the source server. You can use file subscriptions as a convenient way to distribute and update data and application files to the users. For example, it could be an application file that you want to install on the mobile devices, or a resource file required for a Palm OS application to function, or it could be a flat file containing the minutes from a morning meeting.

Perform the following steps in the Mobile Devices Administration Center.

To create a file subscription:

1. Right click on the **Subscriptions** folder and select **Create**. The Create File Subscription window opens.
2. Identify the subscription.
3. Specify the encryption level of a file subscription.
4. Specify the name of the source file.
5. Specify the name of a device type.
6. Specify subscription sets that use the subscription.
7. Click **OK**.

Important: You cannot reference a file to more than one subscription.

Important: If your group of users needs access to several files, you must create several file subscriptions. Assign these subscriptions to a subscription set, then assign the subscription set to the group.

Related tasks

“Creating an object from the object tree” on page 13

This topic describes how to create an object from the object tree.

“Enabling the group to access data and files” on page 26

Bundling subscriptions in a subscription set makes it easier for you to assign a set of subscriptions to multiple groups if necessary. You can assign an unlimited number of subscription sets to a group. To enable the group to access data and files stored on a source system, you assign a subscription set to the group.

“Identifying the file subscription” on page 43

After authenticating a user, the DB2 Everyplace Sync Server determines the group that the user belongs to, then queries the administration control database to determine what subscription is involved in the request. For this reason, the DB2 Everyplace Sync Server must be able to uniquely identify each subscription by name.

“Specifying the encryption level of a file subscription” on page 44

Use this topic to learn the options of the encryption level available.

“Specifying the name of the source file” on page 42

Use this topic to specify the name of the source file.

“Specifying subscription sets that use the subscription” on page 41

After you define a subscription, you must specify the subscription sets that make use of it. When you associate a subscription with a subscription set, you can then assign the subscription set to a group to provide its members with access to the data and files defined in the subscriptions that the subscription set references.

“Flagging a subscription that references a changed file” on page 42

When you make changes to a file referenced in a subscription, you need to flag that the source file has changed so that the DB2 Everyplace Sync Server replicates the file to the subscribed users on the next synchronization.

Administering Custom subscriptions

Custom subscriptions allow you to call stored procedures when you synchronize your mobile device.

To create a Custom subscription:

1. Open the Mobile Devices Administration Center.
2. Right-click on the **Subscriptions** folder in the left pane of the Mobile Devices Administration Center window and select **Create** → **Custom Subscription**. This opens the Create Custom Subscription window.
3. Fill in the following fields:

Name A unique 1 to 18 character name for the subscription

Description

Up to 128 characters of text that describes the subscription

4. Click the **Launch Customizer** button. This opens the Source database window.
 - a. In the **User ID** field, type the DB2 user ID that has access privileges to the target database.
 - b. Type the password for the user ID in the **Password** and **Verify password** fields.
 - c. In the **Other** field, type:

```
dbname=DATABASE;procname=PROCEDURE
```

where

DATABASE

the name of the database used by the stored procedure

PROCEDURE

the name of the stored procedure

Example: To use stored procedure SP1 from data source DS1 and stored procedure SP2 from data source DS2, type the following string:

```
dbname=DS1;procname=SP1;dbname=DS2;procname=SP2
```

5. Click **OK** to close the Source database window. Click **OK** to close the Create Custom Subscription window.

Administering Data Propagator subscriptions

This topic explains the basics of Data Propagator subscriptions and how to create and configure them in DB2 Everyplace.

What is a DataPropagator subscription?: A DataPropagator subscription is a subscription in which the DB2 Everyplace Sync Server replicates tables between a DB2 source database and a DB2 mirror database using DB2 DataPropagator. The DB2 Everyplace Sync Server synchronizes the devices with the mirror database and uses DataPropagator to replicate the mirror database with the source database. A DB2 Everyplace DataPropagator Subscription is mapped to a DataPropagator Subscription Set.

Related tasks

“Batching DB2 Everyplace Sync Server administration commands” on page 90

Creating a DataPropagator table subscription:

This topic provides an overview of the steps that you need to perform to create DataPropagator table subscriptions.

Prerequisite: DB2 Version 9.1 (all platforms) must be installed on both the DB2 Everyplace Sync Server and on the source database.

Note: You cannot use the Mobile Devices Administration Center to create or edit a DataPropagator subscription.

To create a DataPropagator version 8 subscription, you can:

1. Perform manual steps:
 - a. In the DB2 Version 9.1 Replication Center, configure DataPropagator for DB2 Everyplace subscriptions.
 - b. Associate the DataPropagator subscription set with a DB2 Everyplace DataPropagator subscription using the DB2 Everyplace XML scripting tool.
2. Use the XML Scripting Tool to set up DataPropagator replication and DB2 Everyplace Sync Server settings. The XML scripting tool can define a DataPropagator subscription set in the source and the target (mirror) databases and set up the DB2 Everyplace Sync Server tables and triggers in the target (mirror) database. For more information see, DataPropagator subscriptions.

Note: For all processes that use replication, you must start Capture on the source database before the replication begins. Refer to the *DataPropagator Guide and Reference* for more information.

Related tasks

“Configuring DataPropagator for DB2 Everyplace subscriptions”

“Creating a DB2 Everyplace DataPropagator subscription and associating it with a DataPropagator subscription set” on page 48

Related reference

“Data source restrictions for DataPropagator subscriptions” on page 54

Configuring DataPropagator for DB2 Everyplace subscriptions:

Prerequisites

- The source and mirror databases must exist and be cataloged on the machine(s) that run the DB2 Version 9.1 Replication Center and the DB2 Everyplace Mobile Devices Administration Center.
- Because DataPropagator cannot capture remote databases, the mirror database for a DataPropagator subscription must be on the same machine as the replication engine, which is a replication-enabled DB2 Everyplace Sync Server or a command-line process running the dsyreplicate script.
- A DataPropagator subscription cannot use the same mirror database as one used by a JDBC subscription.

The following steps specify DB2 Everyplace requirements for DataPropagator components. Refer to the *DataPropagator Guide and Reference* to ensure that the components are created correctly.

These steps describe how to configure DataPropagator manually using the Replication Center, a procedure that allows you to specify and customize various options. If a simple, default configuration meets your needs, the XML Scripting Tool can perform the task automatically.

Important: If the primary key of a table in a DataPropagator subscription might be updated, you must configure DataPropagator manually by performing the following steps. Do not use XML scripting tool for this because XML scripting tool does not support this function.

To configure DataPropagator for DB2 Everyplace subscriptions:

1. Open the DB2 Version 9.1 Replication Center.
2. Create the Capture Control Tables on the source database.
3. Create the Capture and Apply Control Tables on the mirror database. The capture schema on the mirror database must be ASN.
4. Click **Replicate Definition**. The Manage Target Object Profiles dialog displays.
5. Delete the TG table name prefix and clear the entry field.
6. Enable both the source and mirror Capture Control Server databases for replication. If a database is not already configured with LOGRETAIN=RECOVERY, this property is set, and the Backup Wizard initializes.

7. Register tables in the source database for replication.
 - The **Row Capture Rule** must be to capture changes to all columns.
 - Full refresh of the target table must be allowed.
 - If the source table will be replicated to more than one target table, regardless of whether the target tables are part of a DB2 Everyplace subscription, **Capture changes from replica target table** must be selected.
 - Capture updates as pairs of deletes and inserts should be selected in the following cases:
 - if UPDATES to primary key columns will occur on the source table
 - if data from this source table will be filtered before being sent to the device, and rows which are updated such that they fall out of scope of the filter must be removed from the device.
 - The **Conflict Detection Level** must be Standard detection.
8. Create a DataPropagator subscription set for the mirror database. Use the Set information page of the Create Subscription Set window.
 - In the **Set name** field, type the name of the Subscription Set which must be the name of the DB2 Everyplace DataPropagator Table Subscription.
 - In the **Apply qualifier** field, type the name that you want to use for the apply qualifier.
 - In the **Capture control server alias** field, type the name of the source database.
 - In the **Capture schema** field, click the down arrow and select the schema that you used when Capture was configured.
 - In the **Target server alias** field, type the name of the mirror database.
 - Select the check box next to **Activate the subscription set**.
 - Set the **Data blocking factor** = 0.
 - Select the check box next to **Allow Apply** to use transactional processing for set members.
 - Open the Source-to-target page.
 - Add the members of the subscription set as **Replica** target types.
 - Ensure that the source table primary key is created as the target table primary key in the **Target-Table** Index creation for each table.
 - Select the check box next to **Capture changes** from parent table located on the Replica Definition tab.
9. DB2 Everyplace will initiate the full refresh during the initial replication.

Related tasks

“Creating a DataPropagator table subscription” on page 46

This topic provides an overview of the steps that you need to perform to create DataPropagator table subscriptions.

“Creating a DB2 Everyplace DataPropagator subscription and associating it with a DataPropagator subscription set”

Related reference

“Data source restrictions for DataPropagator subscriptions” on page 54

Creating a DB2 Everyplace DataPropagator subscription and associating it with a DataPropagator subscription set:

Restrictions

- By default, DataPropagator creates mirror tables on its own, non-system managed, table spaces. If you do not specify a table space in the XML to generate the DB2 Everyplace Subscription, the default table space USERSPACE1 is used for the DB2 Everyplace control tables. If you want the DB2 Everyplace control tables associated with the mirror tables to be created in the same table spaces, do one of the following actions:
 - Override the table space creation to specify that the table space is system-managed

- Increase the table space container size to accommodate these control tables (a general recommendation is to increase the container to 5 times the DataPropagator-generated size).
- If Replicate="FALSE" is not specified as an attribute in the AddDPropRSubscription tag, then the mirror database must be local to the machine on which the XML Scripting Tool is executing, and the source capture process must be running.
- DataPropagator source column names cannot be renamed in DataPropagator target (mirror) databases. This is due to a limitation in ASNAPPLY for DataPropagator REPLICA (Update anywhere) subscription sets.

The following section contains an example of an XML Script that associates a DB2 Everyplace DataPropagator Table Subscription with a previously-created DataPropagator Subscription Set by using the DB2 Everyplace XML Scripting Tool.

If CreateDPropRControlTables="TRUE" is specified for AddReplMaster tag, XML Tool will attempt to create the DataPropagator capture control tables in the source database. These are the ASN.IBMSNAP_XXX tables. Similarly if CreateDPropRControlTables="TRUE" is specified for AddReplMirror tag, XML Tool will attempt to create the DataPropagator capture control and apply control tables in the target (mirror) database.

If SetupDPropRSourceAndMirror="TRUE" is specified for AddDPropRSubscription tag, XML Tool will attempt to create the target (mirror) tables, CD tables corresponding to each target table.

These three DataPropagator Subscription-specific flags are FALSE by default (if omitted from the XML). In that case, DataPropagator Replication must have been set up by using the DataPropagator tools prior to running the XML Scripting Tool. The XML Scripting Tool will only create the DB2 Everyplace Sync Server-specific tables and triggers in the mirror database.

Note: The table spaces are being set explicitly to the same as those which were created for the target (mirror) table during the DataPropagator Subscription Set creation. Therefore, the table space creation must be modified.

In this example, the DB2 Everyplace DataPropagator Table Subscription is named DPSUB1 on the Visiting Nurse sample tables.

To create and associate a DataPropagator subscription set with a DB2 Everyplace table subscription:

Tip: If you want to associate the DataPropagator subscription without creating the control tables, set the attribute to false for the following properties:

- CreateDPropRControlTables
- CreateDPropRControlTables
- SetupDPropRSourceAndMirror

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE Root SYSTEM "file:/mdas3.dtd">
```

```
<Root>
  <AddReplMaster CreateDPropRControlTables="TRUE">
    <System>SAD_LPT</System>
    <Instance>DB2</Instance>
    <Database>VNURSE</Database>
    <UserId>db2admin</UserId>
    <Password>db2admin</Password>
    <DatabaseAlias>VNURSE</DatabaseAlias>
  </AddReplMaster>

  <AddReplMirror CreateDPropRControlTables="TRUE">
    <Database>M_VNURSE</Database>
    <UserId>db2admin</UserId>
```

```

    <Password>db2admin</Password>
    <SyncWindow>0</SyncWindow>
</AddReplMirror>

<AddDpropSubscription Replicate="FALSE"
                        SetupDPropSourceAndMirror="TRUE">
    <Name>DPSUB1</Name>
    <SubAdapter>DSYDPROPR</SubAdapter>
    <EncryptionLevel>0</EncryptionLevel>
    <Block>N</Block>
    <MasterSystem>SAD_LPT</MasterSystem>
    <MasterInstance>DB2</MasterInstance>
    <MasterDb>VNURSE</MasterDb>
    <MirrorDb>M_VNURSE</MirrorDb>
    <ApplyQualifier>M_VNURSE</ApplyQualifier>
    <AddTable>
        <TargetSchema>DSYSAMPLE</TargetSchema>
        <TargetTable>VNCONTACT</TargetTable>
        <TableSpace>TSVNCONTACT</TableSpace>
        <SourceSchema>DSYSAMPLE</SourceSchema>
        <SourceTable>VNCONTACT</SourceTable>
        <DeleteAuth>Y</DeleteAuth>
        <InsertAuth>Y</InsertAuth>
        <UpdateAuth>Y</UpdateAuth>
        <SelectAuth>Y</SelectAuth>
        <WhereClauseMirrorToMobile></WhereClauseMirrorToMobile>
        <WhereClauseMasterToMirror></WhereClauseMasterToMirror>
        <JoinFilter>G</JoinFilter>
            <Order>1</Order>
        <Encrypt>N</Encrypt>
        <AddColumn>
            <SourceColumn>PATIENTID</SourceColumn>
            <TargetColumn>PATIENTID</TargetColumn>
            <Subscribe>Y</Subscribe>
            <Replicate>Y</Replicate>
            <Key>Y</Key>
        </AddColumn>
        <AddColumn>
            <SourceColumn>CONTACTID</SourceColumn>
            <TargetColumn>CONTACTID</TargetColumn>
            <Subscribe>Y</Subscribe>
            <Replicate>Y</Replicate>
            <Key>Y</Key>
        </AddColumn>
        <AddColumn>
            <SourceColumn>RELATIONSHIP</SourceColumn>
            <TargetColumn>RELATIONSHIP</TargetColumn>
            <Subscribe>Y</Subscribe>
            <Replicate>Y</Replicate>
            <Key>N</Key>
        </AddColumn>
    </AddTable>
    <AddTable>
        <TargetSchema>DSYSAMPLE</TargetSchema>
        <TargetTable>VNMEDICALRECORD</TargetTable>
        <TableSpace>TSVNMEDICALRECORD</TableSpace>
        <SourceSchema>DSYSAMPLE</SourceSchema>
        <SourceTable>VNMEDICALRECORD</SourceTable>
        <DeleteAuth>Y</DeleteAuth>
        <InsertAuth>Y</InsertAuth>
        <UpdateAuth>Y</UpdateAuth>
        <SelectAuth>Y</SelectAuth>
        <WhereClauseMirrorToMobile></WhereClauseMirrorToMobile>
        <WhereClauseMasterToMirror></WhereClauseMasterToMirror>
        <JoinFilter>G</JoinFilter>
            <Order>2</Order>
        <Encrypt>N</Encrypt>

```

```

<AddColumn>
  <SourceColumn>RECORDID</SourceColumn>
  <TargetColumn>RECORDID</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>Y</Key>
</AddColumn>
<AddColumn>
  <SourceColumn>DATE_C</SourceColumn>
  <TargetColumn>DATE_C</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>N</Key>
</AddColumn>
<AddColumn>
  <SourceColumn>TIME_C</SourceColumn>
  <TargetColumn>TIME_C</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>N</Key>
</AddColumn>
<AddColumn>
  <SourceColumn>PATIENTID</SourceColumn>
  <TargetColumn>PATIENTID</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>N</Key>
</AddColumn>
<AddColumn>
  <SourceColumn>BLOODPRESSURE</SourceColumn>
  <TargetColumn>BLOODPRESSURE</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>N</Key>
</AddColumn>
<AddColumn>
  <SourceColumn>PULSERATE</SourceColumn>
  <TargetColumn>PULSERATE</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>N</Key>
</AddColumn>
<AddColumn>
  <SourceColumn>TEMPERATURE</SourceColumn>
  <TargetColumn>TEMPERATURE</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>N</Key>
</AddColumn>
<AddColumn>
  <SourceColumn>WEIGHT</SourceColumn>
  <TargetColumn>WEIGHT</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>N</Key>
</AddColumn>
<AddColumn>
  <SourceColumn>COMMENT</SourceColumn>
  <TargetColumn>COMMENT</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>N</Key>
</AddColumn>
</AddTable>
<AddTable>
  <TargetSchema>DSYSAMPLE</TargetSchema>
  <TargetTable>VNPERSO</TargetTable>

```

```

<TableSpace>TSVNPERS</TableSpace>
<SourceSchema>DSYSAMPLE</SourceSchema>
<SourceTable>VNPERS</SourceTable>
<DeleteAuth>Y</DeleteAuth>
<InsertAuth>Y</InsertAuth>
<UpdateAuth>Y</UpdateAuth>
<SelectAuth>Y</SelectAuth>
<WhereClauseMirrorToMobile></WhereClauseMirrorToMobile>
<WhereClauseMasterToMirror></WhereClauseMasterToMirror>
<JoinFilter>G</JoinFilter>
<Order>3</Order>
<Encrypt>N</Encrypt>
<AddColumn>
  <SourceColumn>ID</SourceColumn>
  <TargetColumn>ID</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>Y</Key>
</AddColumn>
<AddColumn>
  <SourceColumn>NAME</SourceColumn>
  <TargetColumn>NAME</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>N</Key>
</AddColumn>
<AddColumn>
  <SourceColumn>ADDRESS</SourceColumn>
  <TargetColumn>ADDRESS</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>N</Key>
</AddColumn>
<AddColumn>
  <SourceColumn>CITY</SourceColumn>
  <TargetColumn>CITY</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>N</Key>
</AddColumn>
<AddColumn>
  <SourceColumn>HOMEPHONE</SourceColumn>
  <TargetColumn>HOMEPHONE</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>N</Key>
</AddColumn>
<AddColumn>
  <SourceColumn>WORKPHONE</SourceColumn>
  <TargetColumn>WORKPHONE</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>N</Key>
</AddColumn>
<AddColumn>
  <SourceColumn>MOBILEPHONE</SourceColumn>
  <TargetColumn>MOBILEPHONE</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>N</Key>
</AddColumn>
</AddTable>
<AddTable>
  <TargetSchema>DSYSAMPLE</TargetSchema>
  <TargetTable>VNSCHEDULE</TargetTable>
  <TableSpace>TSVNSCHEDULE</TableSpace>
  <SourceSchema>DSYSAMPLE</SourceSchema>

```



```

<SourceTable>VNSCHEDULE</SourceTable>
<DeleteAuth>Y</DeleteAuth>
<InsertAuth>Y</InsertAuth>
<UpdateAuth>Y</UpdateAuth>
<SelectAuth>Y</SelectAuth>
<WhereClauseMirrorToMobile></WhereClauseMirrorToMobile>
<WhereClauseMasterToMirror></WhereClauseMasterToMirror>
<JoinFilter>G</JoinFilter>
<Order>4</Order>
<Encrypt>N</Encrypt>
<AddColumn>
  <SourceColumn>PATIENTID</SourceColumn>
  <TargetColumn>PATIENTID</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>N</Key>
</AddColumn>
<AddColumn>
  <SourceColumn>TIME_C</SourceColumn>
  <TargetColumn>TIME_C</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>Y</Key>
</AddColumn>
</AddTable>
<AddTable>
  <TargetSchema>DSYSAMPLE</TargetSchema>
  <TargetTable>VNSIGNATURE</TargetTable>
  <TableSpace>TSVNSIGNATURE</TableSpace>
  <SourceSchema>DSYSAMPLE</SourceSchema>
  <SourceTable>VNSIGNATURE</SourceTable>
  <DeleteAuth>Y</DeleteAuth>
  <InsertAuth>Y</InsertAuth>
  <UpdateAuth>Y</UpdateAuth>
  <SelectAuth>Y</SelectAuth>
  <WhereClauseMirrorToMobile></WhereClauseMirrorToMobile>
  <WhereClauseMasterToMirror></WhereClauseMasterToMirror>
  <JoinFilter>G</JoinFilter>
  <Order>5</Order>
  <Encrypt>N</Encrypt>
  <AddColumn>
    <SourceColumn>RECORDID</SourceColumn>
    <TargetColumn>RECORDID</TargetColumn>
    <Subscribe>Y</Subscribe>
    <Replicate>Y</Replicate>
    <Key>Y</Key>
  </AddColumn>
  <AddColumn>
    <SourceColumn>NURSENAME</SourceColumn>
    <TargetColumn>NURSENAME</TargetColumn>
    <Subscribe>Y</Subscribe>
    <Replicate>Y</Replicate>
    <Key>N</Key>
  </AddColumn>
  <AddColumn>
    <SourceColumn>SIGNATURE</SourceColumn>
    <TargetColumn>SIGNATURE</TargetColumn>
    <Subscribe>Y</Subscribe>
    <Replicate>Y</Replicate>
    <Key>N</Key>
  </AddColumn>
</AddTable>
</AddDproprSubscription>
</Root>

```

Related tasks

“Creating a DataPropagator table subscription” on page 46

This topic provides an overview of the steps that you need to perform to create DataPropagator table subscriptions.

“Creating a DB2 Everyplace DataPropagator subscription and associating it with a DataPropagator subscription set” on page 48

Related reference

“Data source restrictions for DataPropagator subscriptions”

Data source restrictions for DataPropagator subscriptions: Restrictions

The following restrictions apply to both Windows and UNIX[®] platforms:

- Before you can apply changes, you must first start the Capture program to capture the changes on the source database.
- Do not create a DataPropagator subscription on tables that have referential-integrity constraints or triggers. Otherwise, replication would fail and would be unrecoverable.
- A mirror database must only replicate with a single source database.
- For z/OS[®] source systems, you must issue a bind Capture job to start the Capture program on the source system. For more information about how to bind the Capture program, see the DB2 Version 9.1 documentation.
- For iSeries:
 - You must start the Capture program on the source database before DB2 Everyplace Sync Server can be started. Use the STRDPRCAP command to start the Capture program.
 - Do not use the XML scripting tool to create control tables for the source database on iSeries because the tool does not support that function. In other words, do not set the **CreateDPropRControlTables** attribute of tag <AddReplMaster> to TRUE. You must create control tables manually by using the CRTDPRTBL command.
 - You must journal source tables manually. Use the STRJRNPf command to journal source tables.

Note: If you use the CREATE SCHEMA statement to create a library that contains the source tables, journaling will perform automatically.

For more information on the commands, see the iSeries Information Center and SQL Replication Guide and Reference.

- The mirror database must be located on the same server as the replication engine, which is a replication-enabled DB2 Everyplace Sync Server or a command-line process running the dsyreplicate script. The DataPropagator Capture program is not able to capture any remote databases.
- By default, DataPropagator will create the mirror tables in their own, non-system managed, table spaces if the subscription is created with Replication Center. The XML Scripting tool does not create the tables in their own non-system managed table spaces. If you want the DB2 Everyplace control tables associated with the mirror tables to be created in the same table spaces, you will need to either override the table space creation to specify that the table space is system-managed, or increase the table space container size to accommodate these control tables (a general recommendation is to increase the container to 5 times the Data Propagator-generated size). If you do not specify a table space in the XML to generate the DB2 Everyplace Subscription, the default table space USERSPACE1 is used for the DB2 Everyplace control tables.
- If Replicate = "FALSE" is not specified as an attribute in the AddDPropRSubscription tag, then the mirror database must be local to the machine on which the XML Scripting Tool is executing, and the source capture process must be running.
- By default, the CommitCount value for a DataPropagator table subscription is 0, which forces all replicating changes to be performed within a single transaction. If an error occurs during replication, all changes are rolled back, and when recovery is attempted, changes are not replayed. However, using this feature increases the amount of transaction space required, especially when an application

generates many changes. Depending on the client environment and application requirements, you might need to set CommitCount to a positive value or increase the transaction log space to support a CommitCount value of 0. You can set the CommitCount value in the XML script that creates or alters the DataPropagator table subscription, or modify it using the DB2 Version 9.1 Replication Center.

- The DB2 Everyplace Sync Server does not synchronize times or timestamps with an hour of 24 correctly due to differences in how time is represented in Java and a data source. A time of "24:00:00", for example, is converted to "00:00:00" and causes the data saved in the mobile database to differ from the data saved in the source database. You should avoid using such an hour in your applications.
- If you are adding a table to a subscription, its source and target schema names, column names, and table names cannot be keywords, reserved words, or special registers in SQL or DB2 Version 9.1.
- A mirror database can replicate only with one source database. DB2 Everyplace does not allow a mirror database to be associated with multiple source databases.
- In rare cases, this can result in an inability to create a table due to non-uniqueness of the table name or column name. In other rare cases it can result in a conversion of a character to multiple characters, as in the German "sharp s" (ß), which will be converted to "SS".

Administering subscription sets

This topic explains how to create and manage subscription sets. A *subscription set* is a container for subscriptions.

What is a subscription set?

A *subscription set* is a container for subscriptions. You assign subscription sets to a group so that its members have access to the data and files defined in the subscriptions that are part of the subscription set. The subscriptions that you assign can be JDBC subscriptions, DataPropagator subscriptions, file subscriptions, or custom subscriptions. Each subscription set can contain any number of subscriptions (limited by mobile device storage capacity).

When users start the synchronization user software on the device, they choose which applications to synchronize. This menu of applications is created from the list of subscription sets associated with the user's group. For more information, see "Allowing multiple devices per user" on page 21.

Related tasks

"Enabling groups to access their data and files" on page 57

"Identifying the subscription set" on page 58

"Specifying what information the subscription set accesses" on page 58

"Making the subscription set available to a group" on page 57

What is a replication subscription?

A *replication subscription*, is any subscription that involves the use of replication between the source and mirror database which resides on the DB2 Everyplace Sync Server. Replication subscriptions include:

- JDBC subscriptions
- DataPropagator subscriptions

Related tasks

"Creating a JDBC subscription" on page 33

Use the **Create JDBC Subscription** window to create a JDBC table subscription. Each table subscription that you create can reference one or more tables.

"Identifying the JDBC subscription" on page 32

Use this topic to learn how to identify the JDBC subscription in the Mobile Devices Administration Center.

"Specifying the encryption level of a JDBC subscription" on page 31

Use this topic to learn the options of the encryption level available.

“Specifying the source database driver and URL” on page 29

You must specify the source and mirror databases for a synchronizing subscription. This topic introduces how to specify the source database.

“Specifying the mirror database driver and URL” on page 29

You must specify the source and mirror databases for a synchronizing subscription. For upload subscriptions, no mirror database is required. This topic introduces how to specify the mirror database.

“Specifying source tables, target tables, and privileges for the subscription” on page 28

“Filtering columns from the source database to the mirror database” on page 63

“Filtering columns from the mirror database to the client device” on page 63

“Filtering rows from the source database to the mirror database” on page 66

“Filtering rows from the mirror database to the client device” on page 67

“Batching replication requests” on page 34

Use this topic to batch replication requests.

“Assigning a subscription set to a JDBC subscription” on page 60

Grouping subscriptions into mobile subscription sets

This task is part of the greater task of Setting up Synchronization. When you have completed the steps below, return to “Synchronization tutorial” on page 144

Administrators use subscription sets to define subscription and user groups. A subscription set can contain several subscriptions. A subscription set links the DB2 Everyplace Sync Server to the DB2 Everyplace Sync Client on a mobile device. The name of a subscription set that is defined on the DB2 Everyplace Sync Server must match the name of the subscription set that is defined for the DB2 Everyplace Sync Client on the mobile device.

To create a subscription set:

Perform the following steps in the Mobile Devices Administration Center.

1. Open the DB2 Control Center and Mobile Devices Administration Center.
 - For Windows, select **Start Programs IBM DB2 Everyplace Start Mobile Devices Administration Center**.
 - For UNIX, change to \$DSYINSTDIR/Server/bin, then run dsyadmin.sh.
2. Right-click on the **Subscription sets** folder in the Mobile Devices Administration Center.
3. Select **Create**.
4. Type a name for the subscription set in the **Name** field.

Note: The name must match the name of a subscription set on the mobile device. This example uses VNurse.

5. Optional: Type a description for the subscription set in the **Description** field. This example uses Nurse Test Subscription set.
6. Click the **Subscriptions** tab.
7. Select the subscriptions you created from the **Available subscriptions** list. This example uses SCHEDULE and CONTACT.
8. Click the > push button to move the subscription to the **Selected subscriptions** list.
9. Click **OK** to close the Create Subscription set window.

The next step is to create mobile users.

Related tasks

“Synchronization tutorial” on page 144

After installing the DB2 Everyplace database and the DB2 Everyplace Sync Server and setting up your device, you are ready to synchronize. The following topics test synchronization using sample data and sample DataPropagator and JDBC subscriptions, but does not use sample upload subscription types.

“Creating a JDBC subscription to access source data” on page 144

“Creating mobile users” on page 22

You need to create a user ID and password for mobile users in order to connect the mobile device to the DB2 Everyplace Sync Server. This topic introduces how you create mobile users in the Mobile Devices Administration Center.

“Creating a data synchronization group” on page 146

“Starting the application server” on page 147

Enabling groups to access their data and files

Perform the following steps in the Mobile Devices Administration Center.

To enable groups to access their data and files:

1. Right-click on the subscription or subscription set to which you want to give your groups access and **Edit**. If you right-clicked on a subscription set, the Change Subscription Set window opens. If you right-clicked on a subscription, the Change Subscription window opens.
2. Identify the subscription set.
3. Specify what information the subscription set accesses.

Tip: Evaluate your current groups of mobile users and determine what information they currently need to perform their work. Assign to the subscription set all subscriptions that represent this information.

4. Make the subscription set available to a group.

Tip: Evaluate which groups require access to the subscriptions referenced in the subscription set. Associate these groups with the subscription set.

5. Click **OK**.

Related tasks

“Identifying the subscription set” on page 58

“Specifying what information the subscription set accesses” on page 58

“Making the subscription set available to a group”

Making the subscription set available to a group:

To allow a group to access subscriptions referenced in a subscription set, you must associate the group with the subscription set.

This task is part of the larger task of enabling groups to access their data and files. When you have completed these steps, return to “Enabling groups to access their data and files.”

1. Go to the Groups tab of the Create Subscription Set or Edit Subscription Set window.
2. In the Available groups list, select the group to which you want to assign the subscription set. To select multiple groups serially, hold down the Ctrl key while selecting; to select multiple groups contiguously, hold down the Shift key while selecting.

If no groups are displayed in the **Available groups** list, you need to create a group. You can click **Create** to open the Create Group window now, or you can create a group later and associate a subscription set with it later. See “Creating a data synchronization group” on page 24 for details on creating a group.

3. Click > to assign the selected groups, or click >> to assign all available groups at once. Assigned groups are displayed in the Selected groups list.

Related tasks

“Enabling groups to access their data and files” on page 57

“Identifying the subscription set”

“Specifying what information the subscription set accesses”

Specifying what information the subscription set accesses:

This task is part of the larger task of enabling groups to access their data and files. When you have completed these steps, return to “Enabling groups to access their data and files” on page 57.

To enable a group of mobile users to access data and files, you must associate the subscription set with replication subscriptions. The replication subscriptions should be the subscriptions that define tables and files as data sources available for replication.

Example: A group of visiting nurses relies on a Palm OS application called vnurse.prc to perform their jobs. The vnurse.prc Palm OS application accesses patient and scheduling information in the PATIENTS and SCHEDULES tables, which reside on a source server. To enable the nurses to use the DB2 Everyplace Sync Server to replicate this data to their user devices, you must define two replication subscriptions:

- One subscription references both of the required tables (PATIENTS and SCHEDULES).
- The other subscription references the resource file for the Palm OS application (vnurse.prc).

After you define the subscriptions, associate the subscriptions with a subscription set. Then, assign that subscription set to the VNURSE group. The visiting nurses can now replicate the data that they need. When the visiting nurses synchronize their data, they receive the latest copy of the resource file, as well as updates to the PATIENTS and SCHEDULES tables.

You can associate an unlimited number of subscriptions with a subscription set.

To associate subscriptions with the subscription set:

Perform the following steps in the Mobile Devices Administration Center.

1. Right-click on an existing subscription set and select **Edit**. The Edit Subscription Set window opens.
2. In the **Available subscriptions** list, select the subscription you want to assign to the subscription set.

To select multiple subscriptions serially, hold down the Ctrl key while selecting; to select multiple subscriptions contiguously, hold down the Shift key.

If no subscriptions are displayed in the **Available subscriptions** list, you will need to create a subscription. You can click **Create** to open the Create JDBC subscription window, Create File subscription window, or Create Custom subscription window now, or you can create a subscription later and associate a subscription set with it later.

3. Click > to assign the selected subscriptions, or click >> to assign all available subscriptions at once. Assigned subscriptions are displayed in the **Selected subscriptions** list.

Related tasks

“Enabling groups to access their data and files” on page 57

“Identifying the subscription set”

“Making the subscription set available to a group” on page 57

Identifying the subscription set:

After authenticating a user, the DB2 Everyplace Sync Server determines the group that the user belongs to, then queries the administration control database to determine the subscriptions that belong to the subscription set referenced in the request. For this reason, the DB2 Everyplace Sync Server must be able to uniquely identify each subscription set by name.

This task is part of the larger task of enabling groups to access their data and files. When you have completed these steps, return to “Enabling groups to access their data and files” on page 57.

To identify the subscription set:

Perform the following steps in the Mobile Devices Administration Center.

1. Click the Identification tab of the Create Upload Subscription or Change Upload Subscription window.
2. Fill in these fields:
 - **Name**
Provide a descriptive, unique name for the subscription set from 1 to 18 characters. You might choose a name that maps to the name of an application already used by a particular group, or you might create a new name here. For example, you might choose VNURSE as the name for the subscription set used by all home health care specialists that work at the hospital.
Subscription set names are case-sensitive.
 - **Description**
Type up to 128 characters of text that describes this subscription set. For example, you might describe the VNURSE subscription set as “Data and files used by all home health care specialists”.

Related tasks

“Enabling groups to access their data and files” on page 57

“Specifying what information the subscription set accesses” on page 58

“Making the subscription set available to a group” on page 57

Assigning a subscription set to a Custom subscription

After you define the subscription, you must specify what subscription sets use it. When you associate a subscription with a subscription set, you can then assign the subscription set to a group to provide members of the group access to the data and files defined in the subscriptions the subscription set references.

This task is part of the larger task of creating a custom subscription. When you have completed these steps, return to “Administering Custom subscriptions” on page 46.

To assign a subscription set to a Custom subscription:

Perform the following steps in the Mobile Devices Administration Center.

1. Go to the subscription sets tab of the Create Custom Subscription window.
2. In the Available subscription sets list, select the subscription set that you want to associate the subscription with. Hold down the Ctrl key while selecting to select multiple subscription sets serially, or hold down the Shift key to select multiple subscription sets contiguously.
3. If no subscription sets appear in the **Available subscription sets** list, click **Create** to enable groups to access their data and files without closing the current Custom Subscription window. The Create Subscription Set window opens.
4. Click > to assign the subscription sets, or click >> to assign all available subscription sets at once. Assigned subscription sets appear in the Selected subscription sets list.

Related tasks

“The default adapters” on page 27

DB2 Everyplace Sync Server provides four adapters that you can use to create subscriptions.

“Creating an upload subscription” on page 39

“Administering Custom subscriptions” on page 46

Custom subscriptions allow you to call stored procedures when you synchronize your mobile device.

Assigning a subscription set to an upload subscription

After you define the subscription, you must specify what subscription sets use it. When you associate a subscription with a subscription set, you can then assign the subscription set to a group to provide members of the group access to the data and files. The data and files that the group can access are defined in the subscriptions that the subscription set references.

One subscription can be used by an unlimited number of subscription sets.

This task is part of the larger task of creating an upload subscription. When you have completed these steps, return to “Creating an upload subscription” on page 39.

To assign a subscription set to an upload subscription:

Perform the following steps in the Mobile Devices Administration Center.

1. Go to the subscription sets tab of the Create Upload Subscription window.
2. In the Available subscription sets list, select the subscription set with which you want to associate the subscription. Hold down the Ctrl key while selecting to select multiple subscription sets serially, or hold down the Shift key to select multiple subscription sets contiguously.
3. If no subscription sets appear in the **Available subscription sets** list, click **Create** to create a subscription set without closing the current Create Upload Subscription window. The Create Subscription Set window opens.
4. Click > to assign the subscription sets, or click >> to assign all available subscription sets at once. Assigned subscription sets appear in the Selected subscription sets list.

Related concepts

“What is an upload subscription?” on page 36

Related tasks

“Creating an upload subscription” on page 39

“Identifying the upload subscription” on page 40

“Specifying the encryption level of an upload subscription” on page 39

“Specifying the source database driver and URL” on page 37

“Specifying source tables and target tables” on page 38

Assigning a subscription set to a JDBC subscription

After you define the subscription, you must specify what subscription sets use it. When you associate a subscription with a subscription set, you can then assign the subscription set to a group to provide members of the group access to the data and files defined in the subscriptions the subscription set references.

One subscription can be used by an unlimited number of subscription sets. However, two subscriptions that are defined with different privileges cannot belong to the same subscription set.

If two subscriptions subscribe to the same source table, each subscription must use a different mirror table. Two subscriptions with different privileges can use the same mirror only if they synchronize with different tables.

This task is part of the larger task of creating a JDBC subscription. When you have completed these steps, return to “Creating a JDBC subscription” on page 33.

To specify the subscription sets that use this subscription:

Perform the following steps in the Mobile Devices Administration Center.

1. Go to the subscription sets tab of the Create JDBC Subscription window.
2. In the **Available subscription sets** list, select the subscription set that you want to associate the subscription with. Hold down the Ctrl key while selecting to select multiple subscription sets serially, or hold down the Shift key to select multiple subscription sets contiguously.

If no subscription sets appear in the **Available subscription sets** list, click **Create** to create a subscription set without closing the current JDBC Subscription window. The Create Subscription Set window opens.

3. Click > to assign the subscription sets, or click >> to assign all available subscription sets at once. Assigned subscription sets appear in the **Selected subscription sets** list.

Related concepts

“Data setup and replication” on page 27

To enable a group of users to access the data and files that its members require, you must define one or more replication subscriptions. A *replication subscription* provides specifications for how the information in a source system (an enterprise server) is to be synchronized with a target system (the mobile device).

Related tasks

“Creating a JDBC subscription” on page 33

Use the **Create JDBC Subscription** window to create a JDBC table subscription. Each table subscription that you create can reference one or more tables.

“Identifying the JDBC subscription” on page 32

Use this topic to learn how to identify the JDBC subscription in the Mobile Devices Administration Center.

“Specifying the encryption level of a JDBC subscription” on page 31

Use this topic to learn the options of the encryption level available.

“Specifying the source database driver and URL” on page 29

You must specify the source and mirror databases for a synchronizing subscription. This topic introduces how to specify the source database.

“Specifying the mirror database driver and URL” on page 29

You must specify the source and mirror databases for a synchronizing subscription. For upload subscriptions, no mirror database is required. This topic introduces how to specify the mirror database.

“Specifying source tables, target tables, and privileges for the subscription” on page 28

“Filtering columns from the source database to the mirror database” on page 63

“Filtering columns from the mirror database to the client device” on page 63

“Filtering rows from the source database to the mirror database” on page 66

“Filtering rows from the mirror database to the client device” on page 67

“Batching replication requests” on page 34

Use this topic to batch replication requests.

Administering table filters in order to manage data

This topic describes how to create and manage filters. A filter allows access to a subset of the data in a database.

What is a data filter?

When users subscribe to a table, they usually only need a subset of table rows or table columns. A *data filter* lets you deliver only the data that a user needs. You can apply horizontal and vertical filters in tandem or separately for specific subscriptions, groups, and users. Data filters only apply to JDBC and DataPropagator subscriptions. There are two different types of data filters: horizontal and vertical.

Horizontal filters filter rows of data. When a client device or the mirror database does not need all the rows of a table, define a horizontal filter to select a subset of rows from the table. There are two different types of horizontal filters: source and mirror. Horizontal source filters filter table rows on the source database to the mirror database. Horizontal mirror filters filter rows on the mirror database to the client device.

Vertical filters filter columns of data. When a client device or mirror database does not need all the columns of a table, define a vertical filter to select a subset of columns from the table. There are two different types of vertical filters: source and mirror. Vertical source filters filter table columns on the source database to the mirror database. Vertical mirror filters filter columns on the mirror database to the client device.

Related concepts

“What is a data synchronization group?” on page 24

A *data synchronization group* is a collection of uniquely named users, or users that have similar data synchronization requirements. Organizing users into groups helps you minimize the effort required to deliver synchronization services to many users in a uniform way.

“What is a subscription set?” on page 55

“Data filter example” on page 75

“What is a user?” on page 18

This topic introduces the definition of *users* and briefly describes its relationship with *groups*.

Related tasks

“Creating a data synchronization group” on page 24

This topic introduces how to create a data synchronization group in the Mobile Devices Administration Center.

“Identifying the group” on page 25

When a user requests synchronization, the DB2 Everyplace Sync Server authenticates the user, and then determines the group to which the user belongs to begin replicating the information referenced in that group’s subscriptions. For this reason, the DB2 Everyplace Sync Server must be able to uniquely identify each group by name.

“Enabling the group to access data and files” on page 26

Bundling subscriptions in a subscription set makes it easier for you to assign a set of subscriptions to multiple groups if necessary. You can assign an unlimited number of subscription sets to a group. To enable the group to access data and files stored on a source system, you assign a subscription set to the group.

“Assigning a user to a group from the Groups window” on page 26

A user can belong to only one group; however, you can move the user from one group to another to accommodate changes in synchronization needs. You can assign an unlimited number of users to a group. This topic introduces how to assign a user to a group from the Groups window.

Filtering data at the subscription level

To filter data at the subscription level:

Perform the following steps in the Mobile Devices Administration Center.

1. Create or edit a JDBC subscription or DataPropagator subscription.
2. Click **Define Subscription**. The Define Replication Subscription window opens.

3. Click **Advanced**. For DB2 DataPropagator subscriptions, the Advanced Subscription Definition window opens. For JDBC subscriptions, the Advanced Replication Subscription window opens.

Note: Horizontal filtering only occurs from server to client. If a client inserts a row that is not be included in the filter, the rows will be inserted into the source.

4. On the rows tab of the subscription window, type an SQL clause in the **All rows needed** box. The entire SQL clause must be typed on one line using the following format:

```
city='myValue'
```

Do not type WHERE at the beginning of the SQL clause. WHERE is automatically appended to the clause typed in the box. For example, the following clause will synchronize only columns that have a city column value equal to Los Angeles:

```
city='Los Angeles, CA 90061'
```

Filtering Data Vertically

This topic explains the uses and types of vertical data filtering in DB2 Everyplace.

Filtering columns from the source database to the mirror database:

Prerequisites:

- Enter values for the required fields (outlined in red) on the Source and Mirror tabs of the Create Subscription or Edit Subscription window.
- The subscription that you want to apply the filter to must have at least one subscribed table.

Use vertical source filters when you do not want the DB2 Everyplace Sync Server to replicate certain columns from a source database to a mirror database during replication.

To filter columns from the source database to the mirror database:

Perform the following steps in the Mobile Devices Administration Center.

1. Create or edit a JDBC subscription or DataPropagator subscription.
2. Click **Define Subscription**. The Define Replication Subscription window opens.
3. Select a table and click **Advanced**. For JDBC subscriptions, the Advanced Replication Subscription window opens. For DataPropagator subscriptions, the Advanced Subscription Definition window opens.
4. Clear the **Replicate** check box for all columns that you do not want the mirror database to replicate.
5. Click **OK**.

Related concepts

“Filter terms defined” on page 73

Related tasks

“Filtering columns from the mirror database to the client device”

Filtering columns from the mirror database to the client device:

Use vertical mirror filters when you do not want a client device to subscribe to certain columns that are replicated to a mirror database. Vertical mirror filters unsubscribe columns from tables on the mirror database during synchronization.

Vertical mirror filters are two-way filters that unsubscribe columns from the mirror database to the client device and from the client device to the mirror database. Because the client device cannot provide values to unsubscribed columns when it inserts rows, you must define a default value in a filter parameter for each unsubscribed column. Filter parameters specify what values the DB2 Everyplace Sync Server should use for unsubscribed columns when a client device inserts rows.

You must define filter parameters at the group level, so that all users in a group have a filter parameter value for each unsubscribed column. However, you can also specify user-specific filter parameter values for some or all users.

To filter columns from the mirror database to the client device:

1. Filter columns from the mirror database to the client device at the subscription level.
2. Define filter parameters for unsubscribed columns at the group level.
3. Optional: If individual users need different filter parameter values for unsubscribed columns, define filter parameters for unsubscribed columns at the user level.

Related reference

“Filter parameter uses and limitations” on page 76

A filter parameter consists of a filter parameter name and a filter parameter value. Filter parameters let you define variables so you can let individual groups or users define their own values for vertical mirror filters and horizontal mirror filters.

Filtering columns from the mirror database to the client device at the subscription level:

Prerequisites:

- Enter values for the required fields (outlined in red) on the Source and Mirror tabs of the Create Subscription or Edit Subscription window.
- The subscription that you want to apply the filter to must have at least one subscribed table.

By filtering columns from the mirror database to the client device, you can eliminate columns that the client device does not need.

This task is part of the larger task of filtering columns from the mirror database to the client device. When you finish these steps, return to “Filtering columns from the mirror database to the client device” on page 63.

To filter columns from the mirror database to the client device at the subscription level:

Perform the following steps in the Mobile Devices Administration Center.

1. Create or edit a JDBC subscription or DataPropagator subscription.
2. Click **Define Subscription**. The Define Replication Subscription window opens.
3. Select a table that you want to filter and click **Advanced**. For JDBC subscriptions, the Advanced Replication Subscription window opens. For DataPropagator subscriptions, the Advanced Subscription Definition window opens.
4. On the Target Columns tab, clear the **Subscribe** check box for all columns that you do not want client devices to subscribe to.
5. For each unsubscribed column, define a filter parameter name:
 - a. Click **Change**.
 - b. In the Change Column window, type the filter parameter name in the **Default value data filter name** field.

Restriction: The parameter must begin with a colon (:). The rest of the name must also follow DB2 Version 9.1 rules for ordinary identifiers. See the DB2 Version 9.1 documentation for more information.

- c. Click **OK**.
 - d. Repeat steps a through c for each unsubscribed column.
6. Click **OK**.

You must give the parameter name a value by setting it in the group or user. You can set a default value for a group to apply to all users in that group, or you can override the group value by setting specific values for each user.

Related tasks

“Filtering columns from the mirror database to the client device” on page 63

“Defining filter parameters for unsubscribed columns at the group level”

“Defining filter parameters for unsubscribed columns at the user level”

Related reference

“Filter parameter uses and limitations” on page 76

A filter parameter consists of a filter parameter name and a filter parameter value. Filter parameters let you define variables so you can let individual groups or users define their own values for vertical mirror filters and horizontal mirror filters.

Defining filter parameters for unsubscribed columns at the group level:

Prerequisites: Before you define filter parameters for a group, perform the following steps:

- Define filter parameter names for each unsubscribed column of this group’s subscriptions.
- Check that the filter parameters that you define in this group are also defined in this group’s subscriptions.
- Open a Create Group or Change Group window.

This task is part of the larger task of filtering columns from the mirror database to the client device. When you finish these steps, return to “Filtering columns from the mirror database to the client device” on page 63.

To define filter parameters for unsubscribed columns at the group level:

Perform the following steps in the Mobile Devices Administration Center.

1. On the Data filter tab of the Create Group or Change Group window, click **Add**.
2. In the **Parameter name** field, type the filter parameter name of an unsubscribed column.
3. In the **Default value** field, type a filter parameter value. The value of \$USERNAME tells the DB2 Everyplace Sync Server to insert the user ID of the user who is updating the row.

Restriction: The parameter must begin with a colon (:). The rest of the name must also follow the conventions that are used by DB2 Version 9.1. See the DB2 Version 9.1 documentation for more information.

4. Click **OK** to close the Add Filter window. To define more filter parameters, click **Apply** and repeat steps 2-3. You must define a filter parameter for each unsubscribed column in this group’s subscriptions.

Related tasks

“Filtering columns from the mirror database to the client device” on page 63

“Filtering columns from the mirror database to the client device at the subscription level” on page 64

“Defining filter parameters for unsubscribed columns at the user level”

Related reference

“Filter parameter uses and limitations” on page 76

A filter parameter consists of a filter parameter name and a filter parameter value. Filter parameters let you define variables so you can let individual groups or users define their own values for vertical mirror filters and horizontal mirror filters.

Defining filter parameters for unsubscribed columns at the user level:

Prerequisites:

- Check if this user is assigned to a group. Users cannot override a filter parameter value if they are not in a group.
- Define filter parameters for this user's group. Users cannot define new filter parameters. They can only override filter parameter values that are already defined in a group.
- Open either a Create User or Change User window.

This task is part of the larger task of filtering columns from the mirror database to the client device. When you finish these steps, return to "Filtering columns from the mirror database to the client device" on page 63.

If individual users need specific values for parameters, you can define a filter parameter value for some or all users. This value overrides the value that you set in the group.

To define filter parameters for unsubscribed columns at the user level:

Perform the following steps in the Mobile Devices Administration Center.

1. On the Data filter tab of the Create Group or Change Group window, select a filter parameter to override.
2. Click **Change**. If the Change button is not available, this user does not belong to a group or this user's group has no defined filter parameters.
3. In the **User Override** field, type the filter parameter value for this user. This value overrides the filter parameter value that you defined for this user's group.
4. Click **OK**.

Related tasks

"Filtering columns from the mirror database to the client device" on page 63

"Filtering columns from the mirror database to the client device at the subscription level" on page 64

"Defining filter parameters for unsubscribed columns at the group level" on page 65

Related reference

"Filter parameter uses and limitations" on page 76

A filter parameter consists of a filter parameter name and a filter parameter value. Filter parameters let you define variables so you can let individual groups or users define their own values for vertical mirror filters and horizontal mirror filters.

Filtering data horizontally

This topic explains the uses and types of horizontal data filtering in DB2 Everyplace.

Filtering rows from the source database to the mirror database:

Prerequisite: Enter values for the required fields (outlined in red) on the Source and Mirror tabs of the Create Subscription or Edit Subscription window.

Use horizontal source filters when you do not want the DB2 Everyplace Sync Server to replicate certain rows from a source database to a mirror database. Horizontal source filters select a subset of rows from tables on the source database during replication.

Horizontal source filters are one-way filters. They do not filter data that the mirror database sends to the source database during replication. They only filter data that comes from the source database to the mirror database during replication.

Horizontal source filters can create the filter scope out-of-sync problem, which causes data inconsistency between the source and mirror under some circumstances. See "Filter terms defined" on page 73 for more information about what the filter scope out-of-sync problem is.

To filter rows from the source database to the mirror database:

Perform the following steps in the Mobile Devices Administration Center.

1. Create or edit a JDBC subscription or DataPropagator subscription.
2. Click **Define Subscription**. The Define Replication Subscription window opens.

Important: If the **Define Subscription** button is disabled, fill in the required fields on the Source and Mirror tabs.

3. Click **Advanced**. For JDBC subscriptions, the Advanced Replication Subscription window opens. For DataPropagator subscriptions, the Advanced Subscription Definition window opens.
4. On the Rows tab, type a search condition in the **All rows needed** text area. Do not include the SQL keyword WHERE. For example, `city = 'San Jose'` and not `WHERE city = 'San Jose'`. Also, do not qualify columns of the domain table. For example, `DSYDOMAIN.city = 'San Jose'`, and `mytable.city = 'San Jose'` are invalid search conditions.

Related concepts

“Filter terms defined” on page 73

Related tasks

“Filtering rows from the mirror database to the client device”

Related reference

“Filter parameter uses and limitations” on page 76

A filter parameter consists of a filter parameter name and a filter parameter value. Filter parameters let you define variables so you can let individual groups or users define their own values for vertical mirror filters and horizontal mirror filters.

Filtering rows from the mirror database to the client device:

Prerequisites:

Read the topics on filter syntax and limitations and filter scope feature uses and limitations.

Use horizontal mirror filters when you do not client devices to subscribe certain rows from a mirror database. A horizontal mirror filter uses an SQL search condition to select a subset of rows from tables on the mirror database during synchronization. You can also use filter parameters in horizontal mirror filters to control which rows the DB2 Everyplace Sync Server sends to specific groups and users.

Horizontal mirror filters are one-way filters. They do not filter data that a client device sends to the mirror database during synchronization. They filter data that comes to the client device from the mirror database during synchronization.

Horizontal source filters can create the filter scope out-of-sync problem, which causes data inconsistency between the source and mirror under some circumstances. See “Filter terms defined” on page 73 for more information about what the filter scope out-of-sync problem is.

The filter scope feature solves the problems that the filter scope out-of-sync problem causes. The filter scope feature causes longer synchronization times, but maintains data integrity. You must decide whether to enable the filter scope feature. The filter scope feature is disabled globally by default. You can change the global setting to apply to all tables, or you can override the global setting on a per-table basis.

To filter rows from the mirror database to the client device:

1. Filter rows from the mirror database to the client device at the subscription level.
2. Optional: If you used filter parameters in the horizontal mirror filter, define filter parameters for groups.

3. Optional: If individual users need to subscribe to different subsets of rows, define filter parameters for users.
4. Optional: Enable the filter scope feature globally.
5. Optional: Enable the filter scope feature on a per-table basis.

Related concepts

“Filter terms defined” on page 73

“Filter scope feature uses and limitations” on page 73

When enabled, the filter scope feature handles the filter scope out-of-sync problem and maintains data integrity. There are some cases, however, that prevent it from working correctly. Find a known list of the limitations of the filter scope feature in this section.

Related tasks

“Filtering rows from the mirror database to the client device at the subscription level”

“Defining horizontal mirror filter parameters at the group level” on page 69

“Defining horizontal filter parameters at the user level” on page 70

“Enabling or disabling the filter scope feature globally” on page 72

“Enabling or disabling the filter scope feature on a per-table basis” on page 72

Related reference

“Filter parameter uses and limitations” on page 76

A filter parameter consists of a filter parameter name and a filter parameter value. Filter parameters let you define variables so you can let individual groups or users define their own values for vertical mirror filters and horizontal mirror filters.

“Horizontal mirror filter syntax” on page 71

DB2 Everyplace filter declarations follow DB2 Version 9.1 syntax for search conditions except for the limitations below. See the DB2 Version 9.1 documentation for the syntax diagram.

Filtering rows from the mirror database to the client device at the subscription level:

Read the topic on filter scope feature uses and limitations before defining any horizontal mirror filters.

Filtering rows from the mirror database to the client device lets you eliminate rows the client device does not need. You must open a Create or Edit JDBC Subscription window or a Create or Edit DataPropagator Subscription window to define a horizontal mirror filter. This task is part of the larger task of filtering columns from the mirror database to the client device. When you finish these steps, return to “Filtering rows from the mirror database to the client device” on page 67.

To filter rows from the mirror database to the client device at the subscription level:

Perform the following steps in the Mobile Devices Administration Center.

1. Click **Define Subscription**. The Define Replication Subscription window opens.

Important: If the **Define Subscription** button is disabled, fill in the required fields on the Source and Mirror tabs.

2. Select a table that you want to filter and click **Advanced**. For JDBC subscriptions, the Advanced Replication Subscription window opens. For DataPropagator subscriptions, the Advanced Subscription Definition window opens.

Important: If the **Advanced** button is disabled, you must first add a table to the subscription.

3. On the Rows tab, type a where-clause in the **Subset of rows for individual users** text area. Do not include the SQL keyword WHERE.
 - You can type a static value to filter for such as `city = 'San Jose'`.

- You can also use filter parameters, which let you filter data at the group or user level. A filter such as `city = :CITY_PARAM` lets you specify specific values for the parameter `:CITY_PARAM` for individual groups and users to this subscription. A parameter must begin with a colon and the rest of the name must follow DB2 Version 9.1 rules for ordinary identifiers as described under in the DB2 Version 9.1 documentation.

Related concepts

“Filter terms defined” on page 73

“Filter scope feature uses and limitations” on page 73

When enabled, the filter scope feature handles the filter scope out-of-sync problem and maintains data integrity. There are some cases, however, that prevent it from working correctly. Find a known list of the limitations of the filter scope feature in this section.

Related tasks

“Defining horizontal mirror filter parameters at the group level”

“Defining horizontal filter parameters at the user level” on page 70

“Enabling or disabling the filter scope feature globally” on page 72

“Enabling or disabling the filter scope feature on a per-table basis” on page 72

Related reference

“Filter parameter uses and limitations” on page 76

A filter parameter consists of a filter parameter name and a filter parameter value. Filter parameters let you define variables so you can let individual groups or users define their own values for vertical mirror filters and horizontal mirror filters.

“Horizontal mirror filter syntax” on page 71

DB2 Everyplace filter declarations follow DB2 Version 9.1 syntax for search conditions except for the limitations below. See the DB2 Version 9.1 documentation for the syntax diagram.

Defining horizontal mirror filter parameters at the group level:

Prerequisite: Before defining group filter parameters, make sure to perform the following steps:

- Define a horizontal mirror filter in the **Subset of rows for individual users text area** text area.
- Check if the subscriptions where you defined horizontal mirror filters are assigned to the group that you are creating or editing.
- Open a Create Group or Change Group window

This task is part of the larger task of filtering rows from the mirror database to the client device. When you finish these steps, return to “Filtering rows from the mirror database to the client device” on page 67.

To define horizontal mirror filter parameters at the group level:

Perform the following steps in the Mobile Devices Administration Center.

1. On the Data filter tab of the Create Group or Change Group window, click **Add**.
2. Type the filter parameter name in the **Parameter name** field. This is the name you used when defining the horizontal mirror filter. It must begin with a colon (:) and follow DB2 Version 9.1 rules for ordinary parameters.
3. Type a default filter parameter value in the **Default value** field. This value is substituted for the parameter name in the horizontal mirror filter. The DB2 Everyplace Sync Server sends all users within this group the rows that fulfill the horizontal mirror filter.
4. Click **OK** to close the Add Filter window. If you have more default filter parameter name and values to define, click **Apply** and repeat steps 2-3.

Related concepts

“Filter terms defined” on page 73

“Filter scope feature uses and limitations” on page 73

When enabled, the filter scope feature handles the filter scope out-of-sync problem and maintains data integrity. There are some cases, however, that prevent it from working correctly. Find a known list of the limitations of the filter scope feature in this section.

Related tasks

“Filtering rows from the mirror database to the client device at the subscription level” on page 68

“Defining horizontal filter parameters at the user level”

“Enabling or disabling the filter scope feature globally” on page 72

“Enabling or disabling the filter scope feature on a per-table basis” on page 72

Related reference

“Filter parameter uses and limitations” on page 76

A filter parameter consists of a filter parameter name and a filter parameter value. Filter parameters let you define variables so you can let individual groups or users define their own values for vertical mirror filters and horizontal mirror filters.

“Horizontal mirror filter syntax” on page 71

DB2 Everyplace filter declarations follow DB2 Version 9.1 syntax for search conditions except for the limitations below. See the DB2 Version 9.1 documentation for the syntax diagram.

Defining horizontal filter parameters at the user level:

Prerequisites: Before you define the filter parameter values for a user, perform the following steps:

- Define filter parameters for this user’s group. Users cannot define new filter parameters. They can only override filter parameter values that are already defined in a group.
- Define a horizontal mirror filter in the **Subset of rows for individual users** text area.
- Open a Create User or Change User window.

This task is part of the larger task of filtering rows from the mirror database to the client device. When you finish these steps, return to “Filtering rows from the mirror database to the client device” on page 67.

To filter rows from the mirror database to the client device at the user level:

Perform the following steps in the Mobile Devices Administration Center.

1. On the Data filter tab of the Create User or Change User window, select a parameter to override.
2. Click **Change...** If the Change button is disabled, this user does not belong to a group or this user’s group has no defined default filter parameter names and values.
3. Type the default filter parameter value that you want this user to have in the **User Override** field. This value overrides the default filter parameter value you defined for this user’s group.
4. Click **OK**.

Related concepts

“Filter terms defined” on page 73

“Filter scope feature uses and limitations” on page 73

When enabled, the filter scope feature handles the filter scope out-of-sync problem and maintains data integrity. There are some cases, however, that prevent it from working correctly. Find a known list of the limitations of the filter scope feature in this section.

Related tasks

“Filtering rows from the mirror database to the client device at the subscription level” on page 68

“Defining horizontal mirror filter parameters at the group level” on page 69

“Enabling or disabling the filter scope feature globally” on page 72

“Enabling or disabling the filter scope feature on a per-table basis” on page 72

Related reference

“Filter parameter uses and limitations” on page 76

A filter parameter consists of a filter parameter name and a filter parameter value. Filter parameters let you define variables so you can let individual groups or users define their own values for vertical mirror filters and horizontal mirror filters.

“Horizontal mirror filter syntax”

DB2 Everyplace filter declarations follow DB2 Version 9.1 syntax for search conditions except for the limitations below. See the DB2 Version 9.1 documentation for the syntax diagram.

Horizontal mirror filter syntax:

DB2 Everyplace filter declarations follow DB2 Version 9.1 syntax for search conditions except for the limitations below. See the DB2 Version 9.1 documentation for the syntax diagram.

Lookup tables

- All lookup tables used be in the same subscription as the domain table.
- A horizontal mirror filter can reference more than one lookup table.
- A horizontal mirror filter can reference the same lookup table more than once.
- A lookup table cannot have its schema name or table name quoted, because names are treated case-insensitively. In other words, delimited identifiers are not supported.

Naming restrictions

- Do not use the SQL keyword WHERE as a correlation name.
- Do not use column names, table names, and correlation names that are identical to a SQL keyword. If you do, explicitly specify the schema name of each lookup table and put a pound sign (#) in front of the schema name. Each lookup table must be in the form of #schema.table.

Wrong: x IN (SELECT c FROM t WHERE WHERE WHERE.y = 1)

Correct: x IN (SELECT c FROM t WHERE1 WHERE WHERE1.y = 1)

Wrong: x IN (SELECT FROM FROM FROM FROM WHERE y = 1)

Correct: x IN (SELECT FROM FROM #DB2ADMIN.FROM FROM WHERE y = 1)

- Do not to use column names and correlation names that are identical to a table name. If you do, explicitly specify the schema name of each lookup table and put a pound sign (#) in front of the schema name. Each lookup table must be in the form of #schema.table

Wrong: x IN (SELECT t1 FROM t1 WHERE y = 1)

Correct: x IN (SELECT t1 FROM #DB2ADMIN.t1 WHERE y = 1)

- Any user-defined SQL identifiers, including schema names, table names, column names, and correlation names, must not start with the string "DSY".
- Do not use the pound sign (#) in any column, table, or schema name.

Parameters

- A parameter name must start with a colon (:), followed by an ordinary identifier as defined in the DB2 Version 9.1 documentation.
- Parameters can only appear where a constant is expected.
- The value of a vertical or horizontal mirror filter parameter must be a constant or a special DB2 Everyplace Sync Server variable such as \$USERNAME. A parameter value cannot contain table names, column names, other user-defined parameters, or any SQL constructs.

Table and column name qualification

- Specify schema names explicitly in the filter, because it reduces the chance of errors and confusion. If you do not specify the schema name of a table, the username is used as the default schema name for the mirror database connection.

- Do not qualify a column with a table name. Use a correlation name instead of the table name. Correlation names cannot be the SQL keyword WHERE. If you need to qualify the domain table, the table on which the filter applies, use the special qualifier DSYDOMAIN.

The filter scope feature: This topic explains the functions of the filter scope feature in DB2 Everyplace.

Enabling or disabling the filter scope feature on a per-table basis:

To enable or disable the filter scope feature on a per-table basis:

- In the Mobile Devices Administration Center:
 1. Create or edit a JDBC or DataPropagator subscription.
 2. In the Identification page, click **Define Subscription...**

Note: If the **Define Subscription...** button is disabled, fill in the required fields on the **Source** and **Mirror** pages.

3. Click on **Add...** if you want to add a new table or **Change...** if you want to change the filter scope feature on an existing table.
 4. Select the check box next to **Override join filter global setting**.
 5. Select **Enable** or **Disable** from the combo box.
- In the XML scripting tool:
 1. Set the **JoinFilter** element to **Y** to enable the filter scope feature, **N** to disable the filter scope feature, or **G** to use the filter scope feature global setting.

Related concepts

“Filter terms defined” on page 73

“Filter scope feature uses and limitations” on page 73

When enabled, the filter scope feature handles the filter scope out-of-sync problem and maintains data integrity. There are some cases, however, that prevent it from working correctly. Find a known list of the limitations of the filter scope feature in this section.

Related tasks

“Enabling or disabling the filter scope feature globally”

Enabling or disabling the filter scope feature globally:

To enable or disable the filter scope feature globally:

On the command line, enter `dsysetproperty DSYGdf1t RowFilter.OutOfScope.Delete = value`. A *value* of 1 enables the filter scope feature. A *value* of 0 disables the filter scope feature.

Important: If you change the global value of `RowFilter.OutOfScope.Delete` from 0 to 1, existing filter scope out-of-sync problems on a client device might remain unless you refresh the client device.

The filter scope feature is now enabled or disabled for all subscribed tables. You can enable or disable this feature on a per-table basis using the XML scripting tool or the Mobile Devices Administration Center.

Related concepts

“Filter terms defined” on page 73

“Filter scope feature uses and limitations” on page 73

When enabled, the filter scope feature handles the filter scope out-of-sync problem and maintains data integrity. There are some cases, however, that prevent it from working correctly. Find a known list of the limitations of the filter scope feature in this section.

Related tasks

“Enabling or disabling the filter scope feature on a per-table basis”

Filter scope feature uses and limitations:

When enabled, the filter scope feature handles the filter scope out-of-sync problem and maintains data integrity. There are some cases, however, that prevent it from working correctly. Find a known list of the limitations of the filter scope feature in this section.

- The filter scope feature resolves the filter scope out-of-sync problem for horizontal mirror filters only. It does not resolve the filter scope out-of-sync problem for horizontal source filters.
- Client devices must refresh to maintain data integrity if you make changes to the horizontal mirror filter by editing the search condition in the **Subset of rows for individual users** text area.
For example, changing a filter from "x > 10" to "y < 5" forces the client device to refresh.
- If a horizontal mirror filter contains attributes or functions that depend on stateful attributes that are not stored in one of the tables referenced by the filter, disable the filter scope feature. If you enable the filter scope feature, client devices might not maintain their data integrity.

A common example of a stateful attribute is a timestamp. Suppose a client device is supposed to contain orders that were made in the last 14 days in the following table:

```
create table orders
( orderid integer not null,
  createdate timestamp,
  amount integer, primary key ( orderid ) );
```

The domain table uses `createdate > (current timestamp - 14 days)` as the horizontal mirror filter.

On the first day of synchronization, the client device contains all rows that were made in the last 14 days. On the next day, any rows that were 14 days old are now 15 days old and should no longer be on the client device. The DB2 Everyplace Sync Server cannot detect and send the deletion of these rows to the client.

- If you change the filter scope feature global setting of `RowFilter.OutOfScope.Delete` from 0 to 1, client devices with existing filter-scope out-of-sync problems must refresh to maintain data integrity.
- Disable the filter scope feature if parameter values change often. If the filter scope feature is enabled, the client device must synchronize before the parameter value change without a replication in between to maintain its data integrity. Otherwise, the client device must fully refresh after the parameter value change to maintain its data integrity.
- Do not create tables in the mirror database.

Related concepts

"Filter terms defined"

Related tasks

"Enabling or disabling the filter scope feature globally" on page 72

"Enabling or disabling the filter scope feature on a per-table basis" on page 72

Related reference

"Horizontal mirror filter syntax" on page 71

DB2 Everyplace filter declarations follow DB2 Version 9.1 syntax for search conditions except for the limitations below. See the DB2 Version 9.1 documentation for the syntax diagram.

"Filter parameter uses and limitations" on page 76

A filter parameter consists of a filter parameter name and a filter parameter value. Filter parameters let you define variables so you can let individual groups or users define their own values for vertical mirror filters and horizontal mirror filters.

Filter terms defined:

Domain table

A table that is subscribed with a filter.

Filter scope

The extent of a filter's domain. The filter's domain is determined by the search condition that you

can define in the **Subset of rows for individual users** (horizontal mirror filter) or **All rows needed** (horizontal source filter) text area of the Advanced Replication window of the Mobile Devices Administration Center.

Filter scope feature

A feature that resolves the filter scope out-of-sync problem for horizontal mirror filters. It does not resolve the filter scope out-of-sync problem for horizontal source filters. Use the `dsysetproperty` tool to enable or disable this feature globally. Use the XML scripting tool or the Mobile Devices Administration Center to override the global setting and enable or disable this feature on a per-table basis.

Filter scope out-of-sync problem

This problem causes tables to have rows that do not fall within its filter scope. This problem potentially occurs when you define a horizontal source filter, a horizontal mirror filter, or both. The filter scope feature corrects this problem for horizontal mirror filters only. Horizontal source filters always experience the filter scope out-of-sync problem.

Three specific cases comprise the filter scope out-of-sync problem:

- Rows that are in the filter scope are updated on a client device and are now out of the filter scope. These rows remain on the client device.

For example, a domain table has the following mirror filter applied to it: `price < 100`. If a client device updates a product's price from 95 dollars to 105 dollars, the updated row is no longer in the filter scope, but it remains on the device.

- Unchanged rows that were once in the filter scope are now out of the filter scope because of a filter scope change since the last synchronization. These rows remain on the user device.

For example, a domain table has the following mirror filter applied to it: `item IN (SELECT product FROM inventory WHERE price < 100)`. The lookup table, `inventory`, is updated to reflect a product's price change from 90 dollars to 100 dollars. This product will not be deleted from the client device even though it is now out of the filter scope.

- Unchanged rows that were once out the filter scope are now in the filter scope because of a filter scope change since the last synchronization. These rows are not deleted from the user device.

For example, a domain table has the following mirror filter applied to it: `item IN (SELECT product FROM inventory WHERE price < 100)`. The lookup table is updated to reflect a product's price change from 100 dollars to 90 dollars. This product will not be inserted into the client device even though it is now in the filter scope.

Horizontal mirror filter

This filter determines which rows are subscribed by the client device. Define this filter in the **Subset of rows for individual users** text field on the Advanced Replication Subscription window in the Mobile Devices Administration Center. The horizontal mirror filter is a one way filter that limits the set of rows the DB2 Everyplace Sync Server sends to the client device. However, it does not limit the set of rows that the client device sends to the DB2 Everyplace Sync Server.

Horizontal source filter

This filter determines which rows are replicated by the mirror database. Define this filter in the **All rows needed** text field on the Advanced Replication Subscription window in the Mobile Devices Administration Center. The horizontal source filter is a one way filter that limits the set of rows the DB2 Everyplace Sync Server replicates from the source database. However, it does not limit the set of rows that the DB2 Everyplace Sync Server sends to the source database.

Lookup table

A table that is referenced by the filter on the domain table

For example, in the filter `item IN (SELECT product FROM inventory WHERE price < 100)`, `inventory` is the lookup table.

Related concepts

“Filter scope feature uses and limitations” on page 73

When enabled, the filter scope feature handles the filter scope out-of-sync problem and maintains data integrity. There are some cases, however, that prevent it from working correctly. Find a known list of the limitations of the filter scope feature in this section.

Related tasks

“Enabling or disabling the filter scope feature globally” on page 72

“Enabling or disabling the filter scope feature on a per-table basis” on page 72

Data filter example

Suppose that you provide database administration services for several hospitals in your state. For administrative purposes, patient statistics are kept in a master table that tracks data from all hospital sites that you administer. Nurses and doctors within each hospital need access to the PATIENTS table, but they need only the rows that contains the data of the patients in their hospital. To handle the needs of each hospital, you can create a data filter that specifies the subset of data that each group can access.

Because only subsets of the table are synchronized, filter parameters help improve performance and provide security for group-specific data.

First of all, you should define a horizontal mirror filter `hospital_name = :Hospital` where `hospital_name` is a column, and `:Hospital` is a filter parameter name.

Next, set a filter parameter for your groups of users. Employees at Eastside hospital might have a filter parameter like the one below. The users in this group will only subscribe to rows that pertain to Eastside hospital.

```
Parameter Name = :Hospital  
Default Value = 'Eastside'
```

A filter parameter for a group of users at Foothill hospital might look like this:

```
Parameter Name = :Hospital  
Default Value = 'Foothill'
```

Create groups to divide your users into logical subgroups because it is easier to administer. You can also define filter parameter values for individual users by overriding the group value for `:Hospital`. Users can only override group parameters and cannot define new ones.

Related concepts

“What is a subscription set?” on page 55

“What is a data filter?” on page 62

“What is a user?” on page 18

This topic introduces the definition of *users* and briefly describes its relationship with *groups*.

Related tasks

“Creating a data synchronization group” on page 24

This topic introduces how to create a data synchronization group in the Mobile Devices Administration Center.

“Identifying the group” on page 25

When a user requests synchronization, the DB2 Everyplace Sync Server authenticates the user, and then determines the group to which the user belongs to begin replicating the information referenced in that group’s subscriptions. For this reason, the DB2 Everyplace Sync Server must be able to uniquely identify each group by name.

“Enabling the group to access data and files” on page 26

Bundling subscriptions in a subscription set makes it easier for you to assign a set of subscriptions to multiple groups if necessary. You can assign an unlimited number of subscription sets to a group. To enable the group to access data and files stored on a source system, you assign a subscription set to the group.

“Assigning a user to a group from the Groups window” on page 26

A user can belong to only one group; however, you can move the user from one group to another to accommodate changes in synchronization needs. You can assign an unlimited number of users to a group. This topic introduces how to assign a user to a group from the Groups window.

Filter parameter uses and limitations

A filter parameter consists of a filter parameter name and a filter parameter value. Filter parameters let you define variables so you can let individual groups or users define their own values for vertical mirror filters and horizontal mirror filters.

Vertical mirror filter parameters

Vertical mirror filters use filter parameters to specify values for unsubscribed columns. Vertical mirror filter parameters let individual groups or users specify what value to use for unsubscribed columns when updating rows. When a client device updates a row with an unsubscribed column, the filter parameter that corresponds with each unsubscribed column is used for the value in that column.

Horizontal mirror filter parameters

Horizontal mirror filters use filter parameters to send only the rows a client device needs. They let individual groups or users specify which rows they need when synchronizing. When a client synchronizes with the mirror database, the DB2 Everyplace Sync Server only sends the rows that fulfill the horizontal mirror filter.

Filter parameter limitations

- A parameter name must start with a colon (:) followed by an ordinary identifier. For more information, see the DB2 Version 9.1 documentation.
- If a filter parameter is a character data type, such as VARCHAR or CHAR, the filter value must be enclosed in single quotes. For example, if a filter parameter such as `city = :CITY_PARAM` is defined in a subscription, the filter value defined in a user or group must be enclosed in single quotes. 'San Jose' is a valid filter value, but San Jose is not.
- In a horizontal mirror filter, parameters can only appear where a value is expected.
- A vertical or horizontal mirror filter parameter value must be a constant or a special DB2 Everyplace Sync Server variable such as \$USERNAME. A parameter value cannot contain table names, column names, other user-defined parameters, or any SQL constructs.

Using custom logic in DB2 Everyplace

This topic explains how to develop, deploy, and enable custom logic. You can add custom logic to a subscription to perform actions at specific points during the synchronization and replication processes.

What is custom logic

You can add custom logic to a subscription to perform actions at specific points during the synchronization and replication processes. For example, custom logic can do the following actions:

- Maintain statistics on synchronization activity.
- Maintain statistics on replication activity.
- Perform primary and foreign key substitution on incoming synchronization data.
- Perform conflict resolution, including the ability to merge column values from different users into a single row or calculate new column values.
- Initiate an external process.

Custom logic can execute before, during, and after the initial handling of input data from the users in any table subscriptions and before, during, and after the replication of data between the mirror and source tables.

Custom logic support is limited in the following cases:

- Custom logic is not supported for devices running a DB2 Everyplace Sync Client version earlier than 8.1.4.
- Custom logic applied to data during replication (**Table replicate** policy) is supported for JDBC table subscriptions only.

Related tasks

“Enabling custom logic policies for the sample application” on page 81

“Using the custom logic sample application” on page 82

Custom Logic terminology

The following list defines terms used to discuss and describe custom logic.

Candidate

A candidate comprises a set of data corresponding to a relational row and an operation (UPDATE, INSERT, DELETE, or NONE) associated with the data. A CLCandidate object represents a candidate and also contains metadata.

Field A field holds a value for a column in a relational row.

Logical Device ID

A logical device ID string is generated by the DB2 Everyplace Sync Server and stored on a device to associate a user with a physical device. A DeviceID object represents a logical device ID.

Record

A record corresponds to a specific primary key value, and will have some number of candidates associated with it. A CLRecord object represents a record and also contains metadata.

Replication

Replication refers to the act of synchronizing the data between one or more mirror tables and the corresponding source tables, and includes the DB2 Everyplace Sync Server processing required to perform this activity and record the results.

Synchronization

Synchronization refers to activity between the DB2 Everyplace Sync Server and a device, from the beginning of the request from the device, and including all responses being sent back to the device. The request can include candidates from the device, and the response can include candidates for the device.

Related concepts

“Custom Logic API” on page 84

“Data type mapping for Custom Logic” on page 83

“Using Custom Logic for conflict resolution” on page 80

Developing custom logic

DB2 Everyplace provides an API for developing custom logic. Reference documentation for the custom logic API is located in the <DSYPATH>\doc\lang\javadoc\CustomLogicdirectory, where <DSYPATH> is the directory where DB2 Everyplace is installed and *lang* is the country code, for example, en_US. In that directory, double-click index.html to display the documentation in a Web browser.

The following figure shows the hierarchy of classes related to custom logic policies.

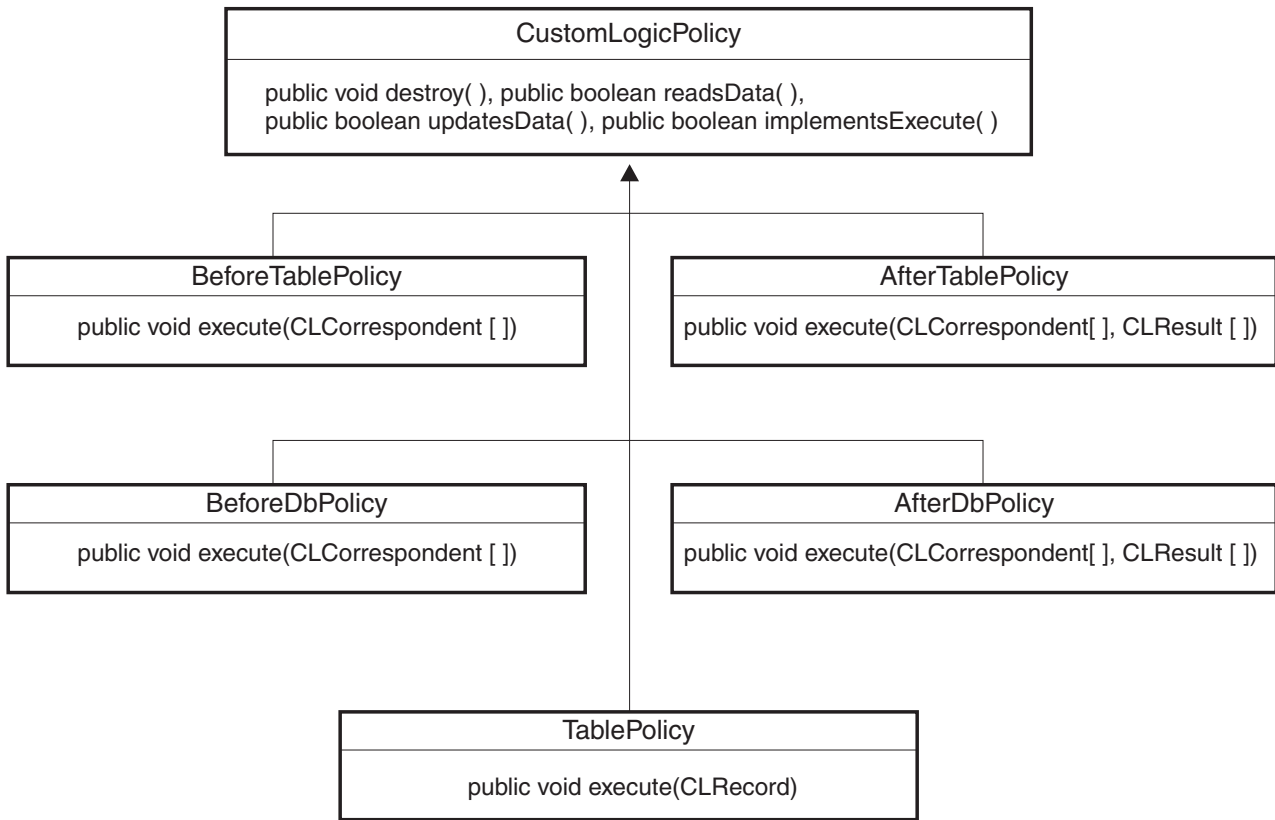


Figure 1. Custom logic class hierarchy.

To develop custom logic:

1. Create the custom logic classes. You can use whatever development environment that you desire to write the Java source files. Place custom logic classes in a package that you create using a naming convention that suits your needs. Because different custom logic might be required for different subscriptions, and possibly even for different tables within the same subscription, ensure uniqueness in the naming convention for the custom logic classes.

2. Compile custom logic classes. Use JDK1.3 to ensure compatibility with the DB2 Everyplace Sync Server. For compilation, the CLASSPATH must include <DSYPATH>\Server\lib\dsvr.jar and <DSYPATH>\Server\lib\dsymd.jar, where <DSYPATH> is the directory where DB2 Everyplace is installed.

3. Apply the custom logic classes to a subscription.

You can use the Mobile Devices Administration Center or the XML Scripting tool to enable policy types and add or change custom logic policies.

The following windows in the Mobile Devices Administration Center have the Custom logic page available:

- Create JDBC subscription
 - Create DataPropagator subscription
 - Change JDBC subscription
 - Change DataPropagator subscription
 - Advanced upload subscription
- a. Start the Mobile Devices Administration Center.
 - b. Create or edit a table subscription.
 - c. Click the **Custom Logic** tab.
 - d. Enable one or more policy types. To enable a policy type:

- 1) Click a name in the **Policy type** column.
 - 2) Click **Change**
 - 3) Type the name of a custom logic class to apply to the policy type.
 - 4) Select **Enabled**.
4. Enable custom logic for specific tables in the subscription.
 - a. Click the **Identification** tab.
 - b. Click **Define subscription**.
 - c. Click a table name in the **Define Replication Subscription** window.
 - d. Click **Advanced** .
 - e. Click the **Custom Logic** tab in the **Advanced Replication Subscription** window.
 - f. Select predefined custom logic policies as desired.

Considerations:

Consider the following points when developing custom logic:

- If the source candidate provided during replication is the default winner, it will prevail if custom logic rejects all candidates.
- The DB2 Everyplace Sync Server applies the prevailing candidate after custom logic decides which candidate should prevail. If the prevailing candidate was originally rejected or was generated by custom logic and the operation of the candidate differs from the operation of the candidate selected by the DB2 Everyplace Sync Server to prevail by default, then referential integrity constraints on the table might be violated.
- The DB2 Everyplace Sync Server processes client changes in autocommit mode when a table policy is associated with a table in a JDBC or a DataPropagator subscription. If the table is in an upload subscription, the DB2 Everyplace Sync Server performs client inserts on all source tables in the subscription in one transaction. If an error occurs during synchronization or replication, the DB2 Everyplace Sync Server can invoke custom logic to reprocess the candidates that were already processed.
- By default, only candidates that intend to perform operations allowed by the administrator are presented to custom logic. To have custom logic decide which operations are allowed, create the table subscription to allow all operations.
- By default, the DB2 Everyplace Sync Server stops the current activity when it receives an exception thrown by custom logic. For example, if custom logic throws an exception to the DB2 Everyplace Sync Server during replication, the replication is left in an intermediate state and will require recovery. During synchronization input data handling, the synchronization session is not completed. To override the default behavior, handle exceptions in custom logic code without throwing them back to the DB2 Everyplace Sync Server.
- Custom logic extending TablePolicy is invoked from within the Replication Engine (JDBC subscriptions only) after the download (source to mirror) takes place, and before the upload (mirror to source). If the custom logic forces a candidate explicitly, the DB2 Everyplace Sync Server, on receiving that indication via the CLRecord, will adjust the input to the upload data for the record such that the JDBC Replication Engine will perform the appropriate distributed transaction on the source and mirror databases. The sync control tables will be modified to reflect the correct replication action.

Deploying custom logic classes

To deploy custom logic classes:

1. Deploy packages containing custom logic classes to directories in the CLASSPATH of each server on which they will run. If the a server cannot find a mapped class, it logs error DSYG013E in DSY.LOG and the affected synchronization or replication will stop.
2. Place the custom logic classes in directories appropriate for the runtime environment, as described below.

Note:

Paths are shown in UNIX syntax. For Windows, use <DSYPATH> instead of \$DSYINSTDIR and back slashes instead of forward slashes.

Invoking replication either as a standalone process or from an administrative action:

\$DSYINSTDIR/Server/classes in subdirectories corresponding to their packages.

Running DB2 Everyplace under WebSphere®:

WebSphere_Home /AppServer/installedApps/ *nodename* /db2everyplace.ear/ IBM_DB2_Everyplace/WEB-INF/classes, where *WebSphere_Home* represents the top-level directory of the WebSphere installation. If the classes directory does not exist, create it.

You can also pack class files in a jar file and place it in the directory *WebSphere_Home* /AppServer/installedApps/ *nodename* /db2everyplace.ear/ IBM_DB2_Everyplace/WEB-INF/lib, where *WebSphere_Home* represents the top-level directory of the WebSphere installation.

Not running DB2 Everyplace under WebSphere:

\$DSYINSTDIR/WAS/installedApps/ *nodename* /db2everyplace.ear/ IBM_DB2_Everyplace.war/ WEB-INF/classes in subdirectories corresponding to their packages. If the classes directory does not exist, create it.

You can also pack class files in a jar file and place it in the directory \$DSYINSTDIR/WAS/ installedApps/ *nodename* /db2everyplace.ear/ IBM_DB2_Everyplace/ WEB-INF/lib.

DB2 Everyplace Enterprise Edition has a class loader that loads classes in the following order:

1. From the installed application’s classes directory
2. From the lib directory inside the deployed application

In addition, within the ../lib path, jar files are prioritized alphabetically. Therefore, deployment location can affect when custom logic classes are loaded. For example, a class stored in both abc.jar and xyz.jar is loaded from abc.jar.

Related concepts

“What is custom logic” on page 76

Related tasks

“Enabling custom logic policies for the sample application” on page 81

“Using the custom logic sample application” on page 82

Using Custom Logic for conflict resolution

The following table describes typical conflict resolution decisions and the actions which should be performed to assert those decisions.

Table 3. Conflict resolution choices and actions

Conflict resolution choice	Action required
Pick a specific candidate to prevail over other candidates	TablePolicy invokes <ul style="list-style-type: none"> • CLRecord.force(CLCandidate), where the passed candidate is the one chosen to prevail in conflict resolution • CLRecord.apply() method, which persists the current state of the CLRecord (if apply is not invoked explicitly, it will be done by the SyncServer on return from the custom logic execute(...) method)

Table 3. Conflict resolution choices and actions (continued)

Conflict resolution choice	Action required
Create a new candidate to prevail over existing candidates (can be a merge of fields from multiple candidates)	<p>TablePolicy invokes</p> <ul style="list-style-type: none"> • CLRecord.createCandidate(...) method <ul style="list-style-type: none"> – This can accept a CLCandidate parameter, in which case a "clone" of that passed candidate will be returned, or – A skeleton CLCandidate can be returned if no CLCandidate parameter is supplied • CLCandidate setter methods on the created candidate for each non-primary key field and operation • CLRecord.force(CLCandidate) method, where the passed candidate is the newly created CLCandidate • CLRecord.apply() method, which persists the current state of the CLRecord (if apply is not invoked explicitly, it will be done by the SyncServer on return from the custom logic execute(...) method)
Accept the default conflict resolution	TablePolicy does nothing or invokes CLRecord.apply()

Related concepts

"Custom Logic terminology" on page 77

"Custom Logic API" on page 84

"Data type mapping for Custom Logic" on page 83

Enabling custom logic policies for the sample application

The Visiting Nurse sample application includes custom logic classes. To use them, you must enable custom logic policies for a subscription and specify a class for each policy.

To enable custom logic policies for the sample application:

1. Start the Mobile Devices Administration Center.
2. Click **Subscriptions** in the object tree.
3. Click **JDBCSUB1** in the contents pane.
4. Click **Selected** → **Edit**.
5. Click the **Custom Logic** tab.
6. Select the **Before database replicate** and **After database replicate** policies. Sample custom logic classes (listed in the **Class name** column) are specified for these policies when the sample application is installed.
7. Click the **Identification** tab.
8. Click **Define subscription**.
9. Click **DSYSAMPLE.VNMEDICALRECORD** in the **Define Replication Subscription 'JDBCSUB1'** window.
10. Click **Advanced** .
11. Click the **Custom Logic** tab in the **Advanced Replication Subscription** window.
12. Select all the following predefined custom logic policies for the table **VNMEDICALRECORD**:
 - **Before table replicate** policy
 - **Table replicate** policy
 - **After table replicate** policy
 - **Before table synchronization input** policy
 - **Table synchronization input** policy

- **After table synchronization input policy**

Related concepts

“What is custom logic” on page 76

Related tasks

“Using the custom logic sample application”

“Deploying custom logic classes” on page 79

Using the custom logic sample application

Prerequisites

- Before you can use custom logic in the sample application, you must enable custom logic policies.
- The following DB2 Everyplace applications and tools are required:
 - DB2 Sync
 - DB2eCLP

The file paths used in this example are Windows-specific and the variable `<DSYPATH>` represents the directory where DB2 Everyplace is installed. For UNIX, replace `<DSYPATH>` with `$DSYINSTDIR`.

To use the custom logic sample application:

Perform the following steps in the Mobile Devices Administration Center.

1. Use DB2 Sync to refresh the initial data on your device (user=nurse1, password=nurse1) .
2. Use DB2eCLP to view the data in your VNMEDICALRECORD table. The schema of this table is:
 - RecordID integer primary key
 - Date_C date
 - Time_C time
 - PatientID char(9) NOT NULL
 - BloodPressure char(7)
 - PulseRate smallint
 - Temperature decimal(4,1)
 - Weight decimal(5,2)
 - Comment varchar(100)
3. Use DB2eCLP to insert a medical record: For example,


```
CLP:> insert into vnmedicalrecord values (1000, current date, current time, '900000001', '140/85', 80, 100.01, 145.50, 'CL statistic');
```
4. Use DB2 Sync to synchronize the VNMEDICALRECORD records with the data source.
 - During synchronization, DB2 Everyplace custom logic writes the log messages to STDOUT if you are using Before Table Synchronization Input, Table Synchronization Input, and After Table Synchronization Input policies. STDOUT is recorded in the file `<DSYPATH>\Server\logs\DB2eServer\native_stdout.log` (search for the strings TableSyncInPolicy, BeforeTableSyncInPolicy, AfterTableSyncInPolicy).
 - Replication occurs periodically every 60 seconds. During replication, custom logic writes the log messages to STDOUT if you are using Before Database Replicate, Before Table Replicate, Table Replicate, After Table Replicate and After Database Replicate policies.
5. Use DB2eCLP to insert a medical record: For example,


```
CLP:> insert into vnmedicalrecord values (-1001, current date, current time, '900000001', '140/85', 80, 100.01, 145.50, 'CL primary key change');
```
6. Use DB2eCLP to review this new record. Note the primary key value is -1001.
7. Use DB2 Sync to synchronize this medical record with the data source. Several operations take place:

- During synchronization, custom logic writes the log messages to the STDOUT showing that it has mapped the primary value -1001 to 1001.
 - When synchronization finishes, use DB2eCLP to view the newly created medical record. Note that the primary key value has been changed to 1001. The sample Custom Logic code that performs this change can be found at: <DSYPATH>\Server\sample\src\com\ibm\mobileservices\sample\VNurseCL\VNurseTableSyncInPolicy.java
8. Wait for the periodic replication. During the replication, custom logic writes the log messages to the STDOUT informing that the new record with the new primary key value 1001 is replicate to the source. When the replication ends, verify that the source contains the new record with the primary key value 1001.
 9. Change periodic replication to replication on demand to see how you can use custom logic for conflict resolution:
 - a. Start the Mobile Devices Administration Center.
 - b. Edit the Visiting Nurse subscription (JDBCSUB1)
 - c. In the Advanced Replication Subscription window, select **Define subscription** and then **Timing**.
 - d. Change 60 to 0.
 10. Insert a medical record in the VNURSE data source as follows:


```
insert into dsysample.vnmedicalrecord (1002, current date, current time, '900000001', '140/85', 80, 100.01, 145.50, 'CL conflict source data');
```
 11. Use DB2eCLP to insert a medical record:


```
CLP:> insert into vnmedicalrecord values (1002, current date, current time, '900000001', '140/85', 80, 100.01, 145.50, 'CL conflict resolution - winner');
```
 12. Use DB2 Sync to synchronize this medical record with the data source.
 13. Perform a manual replication by running the dsyreplicate tool.


```
<DSYPATH>\Server\bin\dsyreplicate.bat M_VN2
```

 During the replication, the Custom Logic writes to the STDOUT informing that it detects there is a conflict and takes the decision to pick the winner from the user (instead of the default rule where the winner is from the source). The sample custom logic code that performs this change can be found at <DSYPATH>\Server\sample\src\com\ibm\mobileservices \sample\VNurseCL\VNurseTableReplicatePolicy.java
 14. Use DB2 Sync to synchronize against the result of the conflict resolution.

Important: because both source and device inserted a conflicting record (with the same primary key 1002), the Custom Logic policy resolves the conflict by selecting the winner from the device "nurse1". This overrode the default conflict resolution algorithm, which always selects the record from the data source as the winner.

15. On the device, use DB2eCLP to see that the winner is the record inserted from the device
16. On the source, verify that the winner is the record inserted from the device.

Related concepts

"What is custom logic" on page 76

Related tasks

"Enabling custom logic policies for the sample application" on page 81

"Deploying custom logic classes" on page 79

Data type mapping for Custom Logic

The getter and setter methods of CLCandidate for the data fields map data types from the java.sql classes to primitive data types as shown in the following table. Custom logic policy implementations must adhere to these mappings when interacting with CLCandidate objects.

Table 4. CLCandidate data type mappings

java.sql Data type	Java Object, primitive data type
java.sql.Types.BIGINT	java.lang.Long, long
java.sql.Types.BINARY	byte[]
java.sql.Types.CHAR	java.lang.String
java.sql.Types.DATE	java.util.Date*
java.sql.Types.DECIMAL	java.math.BigDecimal
java.sql.Types.DOUBLE	java.lang.Double, double
java.sql.Types.FLOAT	java.lang.Double, double
java.sql.Types.INTEGER	java.lang.Integer, int
java.sql.Types.LONGVARBINARY	byte[]
java.sql.Types.LONGVARCHAR	java.lang.String
java.sql.Types.NUMERIC	java.math.BigDecimal
java.sql.Types.REAL	java.lang.Float, float
java.sql.Types.SMALLINT	java.lang.Integer, short
java.sql.Types.TIME	java.util.Date*
java.sql.Types.TIMESTAMP	java.util.Date*
java.sql.Types.TINYINT	java.lang.Integer, byte
java.sql.Types.VARBINARY	byte[]
java.sql.Types.VARCHAR	java.lang.String

* Fields will be presented to the caller of the CLCandidate method `java.util.Date getDate(int)` as the appropriate `java.util.Date` subclass (`java.sql.Date`, `java.sql.Time`, or `java.sql.Timestamp`). If the `java.util.Date` object passed to the CLCandidate method `boolean setField(int, java.util.Date)` for a field of type `java.sql.Types.TIMESTAMP` is not an instance of `java.sql.Timestamp`, resolution will only be to the nearest millisecond. To maintain resolution to the nearest nanosecond, use `java.sql.Timestamp`.

Related concepts

“Custom Logic terminology” on page 77

“Custom Logic API”

“Using Custom Logic for conflict resolution” on page 80

Custom Logic API

The Custom Logic API provides a set of abstract classes which customers can extend to provide custom policies for handling specific events and manipulation of data. Reference documentation for the custom logic API is located in `<DSYPATH>\doc\lang\javadoc\CustomLogic` where `<DSYPATH>` is the root installation directory of DB2 Everyplace and `lang` is the country code (for example, `en_US`). In the `lang` directory, open `index.html` to display the documentation in a Web browser.

Related concepts

“Custom Logic terminology” on page 77

“Data type mapping for Custom Logic” on page 83

“Using Custom Logic for conflict resolution” on page 80

Using the XML scripting tool

This topic describes how to use the XML Scripting tool. The XML Scripting tool automates tasks otherwise performed using the Mobile Devices Administration Center.

About the XML scripting tool

The XML Scripting tool automates tasks otherwise performed using the Mobile Devices Administration Center. For example, in the Mobile Devices Administration Center you can add one user at a time to a group. If you have a large number of users, it is time-consuming to perform this task interactively. With the XML Scripting tool, you can add many users to a group at once. You can also use the XML Scripting tool to copy or move metadata information such as subscriptions, subscription sets, users, groups from one server to several other servers.

In the Mobile Devices Administration Center, you administer synchronization processes through a set of synchronization objects, including users, groups, subscriptions, and subscription sets. When you create, edit, or delete synchronization objects, the Mobile Devices Administration Center records this administration information in the DB2 Everyplace control database (DSYCTLDB). With the XML Scripting tool, you can export information from the control database into an XML script file and edit it for bulk processing or automation.

You can use the XML Scripting tool to process script files from the command line or a Java application. The structure of script files is defined in the XML Scripting tool DTD. Documentation for the Java API is located in the <DSYPATH>\doc\lang\javadoc\xmlAPI where <DSYPATH> is the root installation directory of DB2 Everyplace and *lang* is a language code, for example, en_US.

Related concepts

“XML Scripting tool restrictions” on page 110

Related reference

“Invoking the XML Scripting tool from the command line” on page 86

“Invoking the XML tool programmatically” on page 87

“XML scripting tool DTD” on page 110

Overview of the XML scripting tool

In the Mobile Devices Administrator you can administer synchronization processes through a set of *synchronization objects*. These synchronization objects include users, groups, subscriptions, and subscription sets. When you create, edit, or delete synchronization objects, the Mobile Devices Administrator records this administration information in the DB2 Everyplace control database (DSYCTLDB). What this means is that when you click **OK** on a window in the Mobile Devices Administrator user interface, all the information from that process is stored in the control database.

With the XML Scripting tool, you can export this information from the control database into an XML script file and edit it for bulk processing or automation. For example, in the Mobile Devices Administrator you can add only one user at a time to a group. If you have a large number of users, it would be time consuming to perform this task in the Mobile Devices Administrator. With the XML Scripting tool you can add many users to a group at once. You can also use the XML Scripting tool for easy duplication or movement of the subscription metadata from one server to several other servers.

The XML Scripting tool can be invoked from the command line or it can be invoked programmatically from a Java application.

Related concepts

“About the XML scripting tool”

“XML Scripting tool restrictions” on page 110

Related tasks

“Exporting XML scripts” on page 86

Related reference

“Invoking the XML Scripting tool from the command line” on page 86

“Invoking the XML tool programmatically” on page 87

“XML scripting tool DTD” on page 110

Exporting XML scripts

The XML Scripting tool processes scripts created by exporting data from the control database to XML files.

To export XML scripts:

1. Export the information from the control database into an XML script file using the XML Scripting tool executable `dsyadminxml.bat`. For example, from the `Server/Bin` directory, type:

```
dsyadminxml.bat -x mycurrentsettings.xml
```

where `mycurrentsettings.xml` is the name of your XML script file and `-x` is the command-line parameter that tells the XML scripting tool to create the XML script file and populate it with information from the control database. See “Invoking the XML Scripting tool from the command line” for a complete list of command-line parameters. If the process that you want to automate has *not* already been performed in the Mobile Devices Administration Center, you must perform it before you export the Mobile Devices Administration Center information into an XML script file.

2. Edit the XML script file to manipulate the synchronization objects that you choose. The metadata information in your XML script file will contain several processes that might be in excess of what you need. To identify the processes that you want don’t want, refer to the “XML scripting tool DTD” on page 110.

Related concepts

“About the XML scripting tool” on page 85

“Overview of the XML scripting tool” on page 85

“XML Scripting tool restrictions” on page 110

Related reference

“Invoking the XML Scripting tool from the command line”

“Invoking the XML tool programmatically” on page 87

“XML scripting tool DTD” on page 110

Invoking the XML Scripting tool from the command line

The following table shows the command-line parameters that you can use with the XML Scripting tool.

Table 5. XML Scripting tool parameters

Parameter	Function
<code>-x xmlFile</code>	Exports data from the control database into the XML specified in <code>xmlFile</code> .
<code>-d scriptFile</code>	Applies the XML in the XML script file specified in <code>scriptFile</code> to the control database.
<code>-v scriptFile</code>	Verifies the XML in the XML script file specified in <code>scriptFile</code> .
<code>-?</code>	Displays help.
<code>-x -add</code>	Generates XML that can be used to add the synchronization objects. This is the default behavior of <code>-x</code> .
<code>-x -remove</code>	Generates XML that can be used to remove the synchronization objects.
<code>-x -exclude objectList</code>	Generates XML that excludes the objects specified in <code>objectList</code> , where <code>objectList</code> is a space-separated list of one or more of the following objects: users, groups, subs, subsets, mirrors, masters, adapters, properties.

Table 5. XML Scripting tool parameters (continued)

Parameter	Function
-x -include <i>objectList</i>	Generates XML that includes the objects specified in <i>objectList</i> , where <i>objectList</i> is a space-separated list of one or more of the following objects: users, groups, subs, subsets, mirrors, masters, adapters, properties.
-d <i>scriptFile</i> -v7	Applies the XML in the XML script file specified in <i>scriptFile</i> to the control database using the DB2 Everyplace Version 7 parser.

Examples

```
dsyadminxml -d mynewsettings.xml
```

```
dsyadminxml -x mycurrentsettings.xml
```

```
dsyadminxml -x myfile.xml -include subsets subs
```

Related concepts

“About the XML scripting tool” on page 85

“Overview of the XML scripting tool” on page 85

“XML Scripting tool restrictions” on page 110

Related tasks

“Exporting XML scripts” on page 86

Related reference

“Invoking the XML tool programmatically”

“XML scripting tool DTD” on page 110

Invoking the XML tool programmatically

The following code example shows how to use the XML Scripting tool programmatically from a Java application:

```
package com.ibm.mobileservices.xmlscripting.sample;

/**
 * Title:      XML Scripting tool Sample Application for DB2 Everyplace Sync Server
 * Description: Example of using the XML tool API to write custom administration
                utilities.
 * Copyright:  Copyright (c) 2002
 * Company:   IBM
 */

import java.io.FileWriter;
import java.io.FileReader;

import com.ibm.mobileservices.xmlscripting.XMLTool;
import com.ibm.mobileservices.xmlscripting.XMLToolFactory;

/**
 * Main class for command line invocation of the XML Scripting sample tool.
 */
public class RunXML {

    private RunXML(){}

    /**
     * Method used when running as an application from the command line.
     * @param String[] an array of command line parameters
     * @return void
     */
}
```

```

*
* Command line arguments: -x|-d filename
*
* -x      create XML file from control database.
* -d      apply XML file to control database.
* -v      validate XML file with its DTD.
* filename XML file name to process.
*
* Example:
* java com.ibm.mobileservices.xmlscripting.sample.RunXML -x "c:\mydb.xml"
* Example:
* java com.ibm.mobileservices.xmlscripting.sample.RunXML -d "c:\mycommands.xml"
*/
public static void main(String args[]){
    String filename = null;
    String toOrFromXML = null;
    int xmlType = XMLTool.ADD;

    String xml = null;
    if (args != null && (args.length > 1 && args.length < 3) ){
        toOrFromXML = args[0]; //create xml or db from xml? "-x" means create xml,
        "-d" means populate control db from xml.
        filename = args[1]; //xml file name.
    }

    //check if filename is passed in otherwise display help.
    if (filename == null ||
        filename.equals("") ||
        filename.indexOf("?") > -1 ||
        toOrFromXML.indexOf("?") > -1){
        System.out.println("Command line arguments: -x|-d filename\n"+
            " -x      create XML file from control database.\n"+
            " -d      apply XML file to control database.\n"+
            " -v      validate XML file with its DTD.\n"+
            " filename      XML file name to process."+

            " Example: java com.ibm.mobileservices.xmlscripting.sample.RunXML -x \
            "c:\mydb.xml\"\n"+
            " Example: java com.ibm.mobileservices.xmlscripting.sample.RunXML -d \
            "c:\mycommands.xml\"\n\n");
    }
    return;
}

if (toOrFromXML.equalsIgnoreCase("-x") || toOrFromXML.equalsIgnoreCase("/x"))
    { //generate xml file:
    System.out.println("Generating xml file: " + filename);
    try{
        //backUpOldFile(filename);

        FileWriter fileWriter = null;
        try{
            //get ready to write to the file:
            fileWriter = new FileWriter(filename);

            //create an instance of XMLTool. Each instance opens separate
            //connections to the control database. It is strongly recommended
            //that you use one instance to operate from single thread at a time.
            //If you use the same instance from multiple threads, the threads will
            //typically be synchronized on that database connection. If that causes
            //a bottleneck,create multiple instances of the xml tool to use from
            //multiple threads. Usually that should not be needed.The only case
            //it can become handy is the creation and removalof users from the
            //system.
            //XMLTool xmlTool = XMLToolFactory.createXMLTool();

            //set progress listener
            // (ProgressListener provides hooks to intercept events).

```

```

        //Default progress listener outputs messages to the System.out.
        //A custom progress listener is typically used to re-throw an exception
        //to abort processing of xml tags if the programmer thinks that
        //is necessary.
        xmlTool.setProgressListener
            (XMLToolFactory.createDefaultProgressListener());

        //There are different versions of generateXML, this is one of them:
        xmlTool.generateXML(fileWriter,xmlType);
    }
    finally{
        if (fileWriter!=null) fileWriter.close();
    }

    System.out.println("Done generating xml file: " + filename);

}
catch(Exception ex){
    System.out.println("Failed generating xml file.");
    System.out.println(ex.getMessage());
}
}
else if (toOrFromXML.equalsIgnoreCase("-d") ||
        toOrFromXML.equalsIgnoreCase("/d"))
{ //apply xml to database:
    System.out.println("Applying xml file: " + filename);
    try{
        FileReader aReader = null;
        try{
            //get ready to read the file:
            aReader = new FileReader(filename);

            //create an instance of XML Tool:
            XMLTool xmlTool = XMLToolFactory.createXMLTool();

            //set progress listener:
            xmlTool.setProgressListener
                (XMLToolFactory.createDefaultProgressListener());

            //apply the xml commands:
            xmlTool.applyXML(aReader);
        }
        finally{
            if (aReader!=null) aReader.close();
        }
        System.out.println("Done applying xml.");
    }
    catch(Exception ex){
        System.out.println("Failed applying xml.");
        System.out.println(ex.getMessage());
    }
}
else if (toOrFromXML.equalsIgnoreCase("-v") ||
        toOrFromXML.equalsIgnoreCase("/v"))
{ //validate xml System.out.println("Validating xml file: " + filename);
    try{
        FileReader aReader = null;
        try{
            aReader = new FileReader(filename);
            XMLTool xmlTool = XMLToolFactory.createXMLTool();
            xmlTool.setProgressListener
                (XMLToolFactory.createDefaultProgressListener());
            xmlTool.validateXML(aReader);
        }
        finally{
            if (aReader!=null) aReader.close();
        }
        System.out.println("Done validating xml.");
    }
}
}

```

```

    }
    catch(Exception ex){
        System.out.println("Failed validating xml.");
        System.out.println(ex.getMessage());
    }
}
}
}

```

Related concepts

“About the XML scripting tool” on page 85

“Overview of the XML scripting tool” on page 85

Related tasks

“Exporting XML scripts” on page 86

Related reference

“Invoking the XML Scripting tool from the command line” on page 86

“XML scripting tool DTD” on page 110

Batching DB2 Everyplace Sync Server administration commands

You can use XML Scripting tool to batch DB2 Everyplace Sync Server administration commands. For more information about the XML Scripting tool see, “About the XML scripting tool” on page 85.

For example:

```

<AddReplMirror>
  <Database>M_VNURSE</Database>
  <UserId>db2admin</UserId>
  <Password>db2admin</Password>
  <SyncWindow>0</SyncWindow>
</AddReplMirror>

```

Note: All subscriptions defined against the same mirror database share the same values for batch parameters. Any changes that you make affect all subscriptions defined against the mirror database.

Sample XML scripts

This topic presents the following sample scripts to use with the XML Scripting tool:

- Defining a JDBC subscription
- Removing tables from a JDBC subscription
- Adding tables to a JDBC subscription
- Changing tables of a JDBC Subscription
- Adding and removing indexes for a JDBC subscription
- Defining a DataPropagator subscription
- Creating a group or user
- Changing a user definition
- Defining a subscription set
- Changing a subscription set

Defining a JDBC subscription

This example shows the XML Script to define source (master) database and mirror database information and create a JDBC subscription using these two databases.

The referenced XML DTD file is picked up from the classpath: file:/com/ibm/mobileservices/xmlscripting/mdas3.dtd.

<AddJdbcMaster> tag defines source database connection information.

The JDBC URL, driver, user id and password are required. <AddJdbcMirror> tag defines mirror database connection information. The JDBC URL, driver, user id and password and the sync window (replication frequency) are required.

JDBC master and mirror information must be defined before a JDBC subscription can reference such databases. <AddJdbcSubscription> tag defines a JDBC subscription. When Replicate="TRUE" is specified, replication will be performed to populate the new mirror tables with data from source to get the mirror ready for synchronization with mobile devices. If Replicate="FALSE" is specified, synchronization to this mirror will remain blocked until the first replication of this mirror completes. <AddJdbcSubscription> tag must contain one or more <AddTable> tags. <AddTable> tags should list all the columns of the source table to specify replication and synchronization options for each column using <AddColumn> tag. <SourceSchema> and <SourceTable> identify a table in the source database that we are subscribing to.

<TargetSchema> and <TargetTable> can be anything that will be unique in the mirror database. These names will also be used on the mobile devices. <AddTable> tags can specify indexes for the table columns using <AddIndex> tag. Index names need to be unique for the whole system for the sake of simplicity.

The following example creates a JDBC subscription named 'JDBCSUB_2' which uses VNURSE database as the source and M_VN2 as the mirror. This subscription has only one table that has multiple columns and one ascending index defined (the '+' before the index column means ascending, '-' means descending). Multiple columns can be specified like "+COL1;-COL2;+COL3" for an index.

```
<?xml version="1.0"?>
<!DOCTYPE Root SYSTEM "file:/com/ibm/mobileservices/xmlscripting/mdas3.dtd">
<Root>
  <AddJdbcMaster>
    <Database>jdbc:db2:VNURSE</Database>
    <Driver>COM.ibm.db2.jdbc.app.DB2Driver</Driver>
    <UserId>db2admin</UserId>
    <Password>db2admin</Password>
  </AddJdbcMaster>

  <AddJdbcMirror>
    <Database>jdbc:db2:M_VN2</Database>
    <Driver>COM.ibm.db2.jdbc.app.DB2Driver</Driver>
    <UserId>db2admin</UserId>
    <Password>db2admin</Password>
    <SyncWindow>60</SyncWindow>
  </AddJdbcMirror>

  <AddJdbcSubscription Replicate="TRUE">
    <Name>JDBCSUB_2</Name>
    <SubAdapter>DSYJDBC</SubAdapter>
    <EncryptionLevel>0</EncryptionLevel>
    <Block>N</Block>
    <MasterDb>jdbc:db2:VNURSE</MasterDb>
    <MasterDriver>COM.ibm.db2.jdbc.app.DB2Driver</MasterDriver>
    <MirrorDb>jdbc:db2:M_VN2</MirrorDb>
    <MirrorDriver>COM.ibm.db2.jdbc.app.DB2Driver</MirrorDriver>
    <AddTable>
      <TargetSchema>DSYSAMPLE</TargetSchema>
      <TargetTable>VNPERSO</TargetTable>
      <SourceSchema>DSYSAMPLE</SourceSchema>
      <SourceTable>VNPERSO</SourceTable>
      <DeleteAuth>Y</DeleteAuth>
      <InsertAuth>Y</InsertAuth>
    </AddTable>
  </AddJdbcSubscription>
</Root>
```

```

<UpdateAuth>Y</UpdateAuth>
<SelectAuth>Y</SelectAuth>
<WhereClauseMirrorToMobile>CITY=:CITY</WhereClauseMirrorToMobile>
<WhereClauseMasterToMirror></WhereClauseMasterToMirror>
<JoinFilter>G</JoinFilter>
<Order>1</Order>
<Encrypt>N</Encrypt>
<AddColumn>
  <SourceColumn>ID</SourceColumn>
  <TargetColumn>ID</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>Y</Key>
</AddColumn>
<AddColumn>
  <SourceColumn>NAME</SourceColumn>
  <TargetColumn>NAME</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>N</Key>
</AddColumn>
<AddColumn>
  <SourceColumn>ADDRESS</SourceColumn>
  <TargetColumn>ADDRESS</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>N</Key>
</AddColumn>
<AddColumn>
  <SourceColumn>CITY</SourceColumn>
  <TargetColumn>CITY</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>N</Key>
</AddColumn>
<AddColumn>
  <SourceColumn>HOMEPHONE</SourceColumn>
  <TargetColumn>HOMEPHONE</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>N</Key>
</AddColumn>
<AddColumn>
  <SourceColumn>WORKPHONE</SourceColumn>
  <TargetColumn>WORKPHONE</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>N</Key>
</AddColumn>
<AddColumn>
  <SourceColumn>MOBILEPHONE</SourceColumn>
  <TargetColumn>MOBILEPHONE</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>N</Key>
</AddColumn>
<AddIndex>
  <Name>IDX_CITY</Name>
  <ColNames>+CITY;</ColNames>
</AddIndex>
</AddTable>
</AddJdbcSubscription>

</Root>

```


Removing tables from a JDBC subscription

This example removes two tables from the 'JDBCSUB_2' subscription (assuming the subscription had many tables). Tables of a subscription are identified by their target schema and target table names. When tables are excluded from a subscription, these tables in the mirror database will be dropped. This might cause data loss if these tables still have pending synchronization data that is not yet replicated to the source database.

Replicate="TRUE" attribute tells the XML Scripting tool to replicate existing data before dropping these tables. In some cases, administration might know that there is no data to replicate. Then it is fine to use Replicate="FALSE" to make the script run faster. Also, certain problems that prevent successful replication can be skipped by providing Replicate="FALSE".

```
<?xml version="1.0"?>
<!DOCTYPE Root SYSTEM "file:/com/ibm/mobileservices/xmlscripting/mdas3.dtd">

<Root>
  <AlterJdbcSubscription Replicate="TRUE">
    <Name>JDBCSUB_2</Name>
    <JdbcSubscription>
      <RemoveTable>
        <TargetSchema>DSYSAMPLE</TargetSchema>
        <TargetTable>VNPERSO</TargetTable>
      </RemoveTable>
      <RemoveTable>
        <TargetSchema>DSYSAMPLE</TargetSchema>
        <TargetTable>VNMEDICALRECORD</TargetTable>
      </RemoveTable>
    </JdbcSubscription>
  </AlterJdbcSubscription>
</Root>
```

Adding tables to a JDBC subscription

This example shows how to add tables to an existing subscription. The <Order> tag tells the XML Scripting Tool to make this VNMEDICALRECORD table the second table in this subscription. If there are already more than one table, this change will push other tables to 3rd, 4th, 5th place. The devices that are effected by this change will receive new configuration information and will adjust accordingly. If there are less than 2 tables in this subscription, this table will become the last table.

Replicate="FALSE" flag tells the XML Scripting tool not to replicate this new table after it is created in the mirror. This will cause the mirror table to be empty and there won't be any data to sync. Therefore, the subscription will remain blocked until a successful replication fills this table with data from source and makes it ready for synchronization.

While adding a new table to an existing or a new subscription, all columns of that source table must be listed with their synchronization and replication settings.

```
<?xml version="1.0"?>
<!DOCTYPE Root SYSTEM "file:/com/ibm/mobileservices/xmlscripting/mdas3.dtd">
<Root>
  <AlterJdbcSubscription Replicate="FALSE">
    <Name>JDBCSUB_2</Name>
    <JdbcSubscription>
      <AddTable>
        <TargetSchema>DSYSAMPLE</TargetSchema>
        <TargetTable>VNMEDICALRECORD</TargetTable>
        <SourceSchema>DSYSAMPLE</SourceSchema>
        <SourceTable>VNMEDICALRECORD</SourceTable>
        <DeleteAuth>Y</DeleteAuth>
        <InsertAuth>Y</InsertAuth>
        <UpdateAuth>Y</UpdateAuth>
        <SelectAuth>Y</SelectAuth>
      </AddTable>
    </JdbcSubscription>
  </AlterJdbcSubscription>
</Root>
```

```

<WhereClauseMirrorToMobile></WhereClauseMirrorToMobile>
<WhereClauseMasterToMirror></WhereClauseMasterToMirror>
<JoinFilter>G</JoinFilter>
<Order>2</Order>
<Encrypt>N</Encrypt>
<AddColumn>
  <SourceColumn>RECORDID</SourceColumn>
  <TargetColumn>RECORDID</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>Y</Key>
</AddColumn>
<AddColumn>
  <SourceColumn>DATE_C</SourceColumn>
  <TargetColumn>DATE_C</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>N</Key>
</AddColumn>
<AddColumn>
  <SourceColumn>TIME_C</SourceColumn>
  <TargetColumn>TIME_C</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>N</Key>
</AddColumn>
<AddColumn>
  <SourceColumn>PATIENTID</SourceColumn>
  <TargetColumn>PATIENTID</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>N</Key>
</AddColumn>
<AddColumn>
  <SourceColumn>BLOODPRESSURE</SourceColumn>
  <TargetColumn>BLOODPRESSURE</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>N</Key>
</AddColumn>
<AddColumn>
  <SourceColumn>PULSERATE</SourceColumn>
  <TargetColumn>PULSERATE</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>N</Key>
</AddColumn>
<AddColumn>
  <SourceColumn>TEMPERATURE</SourceColumn>
  <TargetColumn>TEMPERATURE</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>N</Key>
</AddColumn>
<AddColumn>
  <SourceColumn>WEIGHT</SourceColumn>
  <TargetColumn>WEIGHT</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>N</Key>
</AddColumn>
<AddColumn>
  <SourceColumn>COMMENT</SourceColumn>
  <TargetColumn>COMMENT</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>N</Key>

```

```

        </AddColumn>
    </AddTable>
</JdbcSubscription>
</AlterJdbcSubscription>
</Root>

```

Changing tables of a JDBC subscription

This example shows how to change the details of existing tables of existing subscriptions: This script modifies the WHERE clauses that do horizontal filtering for replication and synchronization. Because Replicate="TRUE" is specified, the existing mirror will be replicated to the source database before any change is made. Then changes will be applied and another replication will be attempted. This second replication might fail if you have an error in the WHERE clause. You will then need to revert your changes with another 'AlterJdbcSubscription' tag with Replicate="FALSE" flag. You can also replicate manually using command line utility dsyreplicate.bat after your changes are applied.

```

<?xml version="1.0"?>
<!DOCTYPE Root SYSTEM "file:/com/ibm/mobileservices/xmlscripting/mdas3.dtd">
<Root>
  <AlterJdbcSubscription Replicate="TRUE">
    <Name>JDBCSUB_2</Name>
    <JdbcSubscription>
      <AlterTable>
        <TargetSchema>DSYSAMPLE</TargetSchema>
        <TargetTable>VNPERSO</TargetTable>
        <Table>
          <WhereClauseMirrorToMobile>NAME like :PREFIX</WhereClauseMirrorToMobile>
          <WhereClauseMasterToMirror>NAME like 'A%'</WhereClauseMasterToMirror>
        </Table>
      </AlterTable>
    </JdbcSubscription>
  </AlterJdbcSubscription>
</Root>

```

Adding and removing indexes for a JDBC subscription

This example adds a new index (IDX1_a) to an existing table and removes an existing index (IDX1_b) from this table. Because MirrorOnly="TRUE" is not specified for the index, this index will also be created on target devices which subscribe to this subscription as well.

```

<?xml version="1.0"?>
<!DOCTYPE Root SYSTEM "file:/com/ibm/mobileservices/xmlscripting/mdas3.dtd">
<Root>
  <AlterJdbcSubscription Replicate="FALSE">
    <Name>JDBCSUB_2</Name>
    <JdbcSubscription>
      <AlterTable>
        <TargetSchema>DSYSAMPLE</TargetSchema>
        <TargetTable>VNPERSO</TargetTable>
        <Table>
          <AddIndex>
            <Name>IDX1_a</Name>
            <ColNames>+NAME;</ColNames>
          </AddIndex>
          <RemoveIndex>
            <Name>IDX1_b</Name>
          </RemoveIndex>
        </Table>
      </AlterTable>
    </JdbcSubscription>
  </AlterJdbcSubscription>
</Root>

```

Defining a DataPropagator subscription

<AddReplMaster CreateDPropRControlTables="TRUE"> tag defines source database connection information. CreateDPropRControlTables attribute is optional. It is "FALSE" by default. If it is specified as "TRUE", XML Tool will attempt to create DataPropagator capture control tables in the source database. These are the ASN.IBMSNAP_XXX tables required by Data Propagator.

The system, instance, database and password are required.

<AddReplMirror CreateDPropRControlTables="TRUE"> tag defines mirror database connection information.

The database, user id, password and the sync window (replication frequency) are required.

CreateDPropRControlTables attribute is optional. It is "FALSE" by default. If it is specified as "TRUE", XML Tool will attempt to create DataPropagator capture control and apply control tables in the target (mirror) database. These are the ASN.IBMSNAP_XXX tables required by DataPropagator.

Replication master and mirror information must be defined before a DataPropagator subscription can reference such databases. The control table creation options can be left as "FALSE" if control tables already exist.

<AddDpropRSubscription SetupDPropRSourceAndMirror="TRUE"> tag defines a DPropR subscription for DB2 Everyplace Sync Server. SetupDPropRSourceAndMirror attribute is optional. It is "FALSE" by default. If SetupDPropRSourceAndMirror="FALSE", the replication mechanism between the source and the mirror database must already be setup using DataPropagator tools as if XML Tool or DB2 Everyplace Sync Server never existed. XML Tool picks up from the point where DB2 Everyplace Sync Server specific setup is needed. Refer to DataPropagator documentation to find out how to setup replication sources and a replication subscription to replicate your source and mirror. The mirror tables must already be created before you can use XML Tool to introduce your DataPropagator replication system to the DB2 Everyplace Sync Server through the DataPropagator subscription that you are creating here.

When SetupDPropRSourceAndMirror="TRUE" is specified, XML Tool will attempt to create the target tables, and the CD tables and will populate the DataPropagator control tables (ASN.IBMSNAP_XXX tables) in the source and target databases for you. In this case, you do not need to go through the manual steps of creating DataPropagator subscription sets. XML Scripting Tool will do it for you.

<AddDPropRSubscription> tag must contain one or more <AddTable> tags. <AddTable> tags should list all the columns of the mirror table to specify replication and synchronization options for each column using <AddColumn> tag. Source column names cannot be renamed in target databases due to ASNAPPLY restrictions in DataPropagator REPLICA (Update Anywhere) Subscriptions.

<SourceSchema> and <SourceTable> identify a table in the source database that we are subscribing to. <TargetSchema> and <TargetTable> identify a table in the mirror database that we are subscribing to. These names will also be used on the mobile devices. <AddTable> tags can specify indexes for the table columns using <AddIndex> tag. Index names are unique for the whole system for the sake of simplicity.

By default, the CommitCount value for a DataPropagator table subscription is 0, which forces all replicating changes to be performed within a single transaction. If an error occurs during replication, all changes are rolled back, and when recovery is attempted, changes are not replayed. However, using this feature increases the amount of transaction space required, especially when an application generates many changes. Depending on the client environment and application requirements, you need to set CommitCount to a positive value or increase the transaction log space to support a CommitCount value of 0.

The following example creates a DPropR subscription named 'DPSUB1' which uses VNURSE database as the source and M_VNURSE as the mirror. This subscription has only 1 table that has multiple columns and 1 ascending index defined (the '+' before the index column means ascending, '-' means descending). Multiple columns can be specified like "+COL1;-COL2;+COL3" for an index.

```
<?xml version="1.0"?>
<!DOCTYPE Root SYSTEM "file:/com/ibm/mobileservices/xmlscripting/mdas3.dtd">
<Root>
  <AddReplMaster CreatedPropRControlTables="TRUE">
    <System>SAD_LPT</System>
    <Instance>DB2</Instance>
    <Database>VNURSE</Database>
    <UserId>db2admin</UserId>
    <Password>db2admin</Password>
    <DatabaseAlias>VNURSE</DatabaseAlias>
  </AddReplMaster>

  <AddReplMirror CreatedPropRControlTables="TRUE">
    <Database>M_VNURSE</Database>
    <UserId>db2admin</UserId>
    <Password>db2admin</Password>
    <SyncWindow>3600</SyncWindow>
  </AddReplMirror>

  <AddDpropRSubscription Replicate="FALSE"
    SetupDPropRSourceAndMirror="TRUE">
    <Name>DPSUB1</Name>
    <SubAdapter>DSYDPROPR</SubAdapter>
    <EncryptionLevel>0</EncryptionLevel>
    <Block>N</Block>
    <MasterSystem>SAD_LPT</MasterSystem>
    <MasterInstance>DB2</MasterInstance>
    <MasterDb>VNURSE</MasterDb>
    <MirrorDb>M_VNURSE</MirrorDb>
    <CommitCount>1</CommitCount>
    <AddTable>
      <TargetSchema>DSYSAMPLE</TargetSchema>
      <TargetTable>VNMEDICALRECORD</TargetTable>
      <SourceSchema>DSYSAMPLE</SourceSchema>
      <SourceTable>VNMEDICALRECORD</SourceTable>
      <DeleteAuth>Y</DeleteAuth>
      <InsertAuth>Y</InsertAuth>
      <UpdateAuth>Y</UpdateAuth>
      <SelectAuth>Y</SelectAuth>
      <WhereClauseMirrorToMobile></WhereClauseMirrorToMobile>
      <WhereClauseMasterToMirror></WhereClauseMasterToMirror>
      <JoinFilter>G</JoinFilter>
      <Order>1</Order>
      <Encrypt>N</Encrypt>
      <AddColumn>
        <SourceColumn>RECORDID</SourceColumn>
        <TargetColumn>RECORDID</TargetColumn>
        <Subscribe>Y</Subscribe>
        <Replicate>Y</Replicate>
        <Key>Y</Key>
      </AddColumn>
      <AddColumn>
        <SourceColumn>DATE_C</SourceColumn>
        <TargetColumn>DATE_C</TargetColumn>
        <Subscribe>Y</Subscribe>
        <Replicate>Y</Replicate>
        <Key>N</Key>
      </AddColumn>
      <AddColumn>
        <SourceColumn>TIME_C</SourceColumn>
        <TargetColumn>TIME_C</TargetColumn>
        <Subscribe>Y</Subscribe>
      </AddColumn>
    </AddTable>
  </AddDpropRSubscription>
</Root>
```

```

        <Replicate>Y</Replicate>
        <Key>N</Key>
    </AddColumn>
</AddColumn>
<AddColumn>
    <SourceColumn>PATIENTID</SourceColumn>
    <TargetColumn>PATIENTID</TargetColumn>
    <Subscribe>Y</Subscribe>
    <Replicate>Y</Replicate>
    <Key>N</Key>
</AddColumn>
<AddColumn>
    <SourceColumn>BLOODPRESSURE</SourceColumn>
    <TargetColumn>BLOODPRESSURE</TargetColumn>
    <Subscribe>Y</Subscribe>
    <Replicate>Y</Replicate>
    <Key>N</Key>
</AddColumn>
<AddColumn>
    <SourceColumn>PULSERATE</SourceColumn>
    <TargetColumn>PULSERATE</TargetColumn>
    <Subscribe>Y</Subscribe>
    <Replicate>Y</Replicate>
    <Key>N</Key>
</AddColumn>
<AddColumn>
    <SourceColumn>TEMPERATURE</SourceColumn>
    <TargetColumn>TEMPERATURE</TargetColumn>
    <Subscribe>Y</Subscribe>
    <Replicate>Y</Replicate>
    <Key>N</Key>
</AddColumn>
<AddColumn>
    <SourceColumn>WEIGHT</SourceColumn>
    <TargetColumn>WEIGHT</TargetColumn>
    <Subscribe>Y</Subscribe>
    <Replicate>Y</Replicate>
    <Key>N</Key>
</AddColumn>
<AddColumn>
    <SourceColumn>COMMENT</SourceColumn>
    <TargetColumn>COMMENT</TargetColumn>
    <Subscribe>Y</Subscribe>
    <Replicate>Y</Replicate>
    <Key>N</Key>
</AddColumn>
</AddTable>
</AddDproprSubscription>
</Root>

```

Creating a group or user

<AddGroup> tag defines a Group for DB2 Everyplace Sync Server. Users that can synchronize data with DB2 Everyplace Sync Server must be associated with existing Groups. The Group is identified by its name. It can be enabled or disabled for synchronization. A Group can have 1 or more data filters defined using <AddDataFilter> tag. These data filters are used in where clauses of subscribed tables to do horizontal filtering.

For example, a Group can have a data filter named ':CITY'. The WHERE clause of a table specify a condition like MYSCHEMA.MYTABLE.CITYCOLUMN=:CITY. Then, in case of this example, if we were to create this Group and synchronize its users, we would be selecting rows where MYSCHEMA.MYTABLE.CITYCOLUMN='SAN JOSE' by default. Note that, individual users can override a data filter of their Group to specify a different data filter value. A user cannot override a data filter that its Group does not provide.

In the example below, the User 'nurse_2' is created and assigned to 'GROUP_2'. It overrides data filter :CITY to provide a different city name specific to this 'nurse_2'. However, this user does not override the data filter ':COUNTRY'. It inherits the ':COUNTRY' data filter from its group 'GROUP_2' with its value 'USA'. Lastly, an existing user named 'nurse_3' is assigned to this new group 'GROUP_2'.

Note: Changing the group of a user causes the user's devices to be reset. That is because the subscription schema of one group not match the other. DO NOT change the groups of your users unless you are prepared for potential data loss on the mobile devices

```
<?xml version="1.0"?>
<!DOCTYPE Root SYSTEM "file:/com/ibm/mobileservices/xmlscripting/mdas3.dtd">

<Root>
  <AddGroup>
    <Name>GROUP_2</Name>
    <Description>A test group</Description>
    <Enabled>Y</Enabled>
    <AddDataFilter>
      <Name>:CITY</Name>
      <Value>'SAN JOSE'</Value>
    </AddDataFilter>
    <AddDataFilter>
      <Name>:COUNTRY</Name>
      <Value>'USA'</Value>
    </AddDataFilter>
    <IncludeSubscriptionSet>
      <Name>SUBSET_2</Name>
    </IncludeSubscriptionSet>
  </AddGroup>

  <AddUser>
    <Name>nurse_2</Name>
    <UserGroup>GROUP_2</UserGroup>
    <Description>A test user</Description>
    <Password>nurse_2</Password>
    <OverrideDataFilter>
      <Name>:CITY</Name>
      <Value>SANTA CLARA</Value>
    </OverrideDataFilter>
  </AddUser>

  <AlterUser>
    <Name>nurse_3</Name>
    <User>
      <UserGroup>GROUP_2</UserGroup>
    </User>
  </AlterUser>
</Root>
```

Changing a user definition

This script changes the description of a user named nurse_5, sets a new password, and overrides a data filter :DF1 that the user inherits from its assigned Group, whatever that might be. If the current group of the user does not have a data filter called :DF1, this script will fail and no changes will be applied for this <AlterUser> tag.

```
<?xml version="1.0"?>
<!DOCTYPE Root SYSTEM "file:/com/ibm/mobileservices/xmlscripting/mdas3.dtd">

<Root>
  <AlterUser>
    <Name>nurse_5</Name>
    <User>
      <Description>A new description.</Description>
      <Password>nurse5pass</Password>
```

```

    <OverrideDataFilter>
      <Name>:DF1</Name>
      <Value>-1</Value>
    </OverrideDataFilter>
  </User>
</AlterUser>
</Root>

```

Defining a subscription set

<AddSubscriptionSet> tag defines a new Subscription Set for synchronization purposes. <IncludeSubscription> tag is used to associate existing subscriptions with this subscription set. In this example, contents of Subscription Set SUBSET_2 are JDBC SUB_2 and FILE SUB_3 where this example assumes that JDBC SUB_2 and FILE SUB_3 are existing subscriptions.

The order of the subscriptions inside a subscription set is by default the order that they appear in the XML. In this example, JDBC SUB_2 will be synchronized before FILE SUB_3. <Order> tag can be specified inside <IncludeSubscription> tag to explicitly specify relative ordering.

```

<?xml version="1.0"?>
<!DOCTYPE Root SYSTEM "file:/com/ibm/mobileservices/xmlscripting/mdas3.dtd">
<Root>
  <AddSubscriptionSet>
    <Name>SUBSET_2</Name>
    <IncludeSubscription>
      <Name>JDBC SUB_2</Name>
    </IncludeSubscription>
    <IncludeSubscription>
      <Name>FILE SUB_3</Name>
    </IncludeSubscription>
  </AddSubscriptionSet>
</Root>

```

Changing a subscription set

This example adds a subscription named 'JDBC SUB_4' (assuming JDBC SUB_4 already exists) to the subscription set 'SUBSET_2' and makes this new subscription the first subscription in this subscription set.

```

<?xml version="1.0"?>
<!DOCTYPE Root SYSTEM "file:/com/ibm/mobileservices/xmlscripting/mdas3.dtd">
<Root>
  <AlterSubscriptionSet>
    <Name>SUBSET_2</Name>
    <SubscriptionSet>
      <IncludeSubscription>
        <Name>JDBC SUB_4</Name>
        <Order>1</Order>
      </IncludeSubscription>
    </SubscriptionSet>
  </AlterSubscriptionSet>
</Root>

```

Related concepts

"About the XML scripting tool" on page 85

"Overview of the XML scripting tool" on page 85

Related tasks

"Exporting XML scripts" on page 86

Related reference

"Invoking the XML Scripting tool from the command line" on page 86

"XML scripting tool DTD" on page 110

JDBC subscriptions: This example shows the XML Script to define source (master) database and mirror database information and create a JDBC subscription using these two databases.

The referenced XML DTD file is normally picked up from the classpath: file:/com/ibm/mobileservices/xmlscripting/mdas3.dtd.

<AddJdbcMaster> tag defines source database connection information. The JDBC URL, driver, user id and password are required. <AddJdbcMirror> tag defines mirror database connection information. The JDBC URL, driver, user id and password and the sync window (replication frequency) are required.

JDBC source and mirror information must be defined before a JDBC subscription can reference such databases.

<AddJdbcSubscription> tag defines a JDBC subscription. When Replicate="TRUE" is specified, replication will be performed to populate the new mirror tables with data from source to get the mirror ready for synchronization with mobile devices. If Replicate="FALSE" is specified, synchronization to this mirror will remain blocked until the first replication of this mirror completes. <AddJdbcSubscription> tag must contain 1 or more <AddTable> tags. <AddTable> tags should list all the columns of the source table to specify replication and synchronization options for each column using <AddColumn> tag. <SourceSchema> and <SourceTable> identify a table in the source database that we are subscribing to.

<TargetSchema> and <TargetTable> can be anything that will be unique in the mirror database. These names will also be used on the mobile devices. <AddTable> tags can specify indexes for the table columns using <AddIndex> tag. Index names need to be unique for the whole system for the sake of simplicity.

Horizontal mirror filters defined with the <WhereClauseMirrorToMobile> tag in DB2 Everyplace 8.1.4 referenced source database column names, which made it hard to support different source databases because of varying degrees of column name compatibility. DB2 Everyplace 8.2 now references mobile database column names instead of source database column names in horizontal mirror filters to better support different source databases. Only new subscriptions use mobile database column names. Migrated subscriptions continue to use source column names in the mirror database. If you are exporting a subscription to XML from DB2 Everyplace 8.1.4, and want to use it to create a new subscription, modify the <WhereClauseMirrorToMobile> tag to reference mobile database target names. Subscriptions that do not have horizontal mirror filters or have the same names for source database and mobile database columns do not need to do this.

The example below creates a JDBC subscription named 'JDBCSUB_2' that uses VNURSE database as the source and M_VN2 as the mirror. This subscription has only 1 table that has multiple columns and 1 ascending index defined (the '+' before the index column means ascending, '-' means descending). Multiple columns can be specified like "+COL1;-COL2;+COL3" for an index.

```
<?xml version="1.0"?>
<!DOCTYPE Root SYSTEM "file:/com/ibm/mobileservices/xmlscripting/mdas3.dtd">
<Root>
  <AddJdbcMaster>
    <Database>jdbc:db2:VNURSE</Database>
    <Driver>COM.ibm.db2.jdbc.app.DB2Driver</Driver>
    <UserId>db2admin</UserId>
    <Password>db2admin</Password>
  </AddJdbcMaster>

  <AddJdbcMirror>
    <Database>jdbc:db2:M_VN2</Database>
    <Driver>COM.ibm.db2.jdbc.app.DB2Driver</Driver>
    <UserId>db2admin</UserId>
    <Password>db2admin</Password>
    <SyncWindow>60</SyncWindow>
  </AddJdbcMirror>

  <AddJdbcSubscription Replicate="TRUE">
```

```

<Name>JDBCSUB_2</Name>
<SubAdapter>DSYJDBC</SubAdapter>
<EncryptionLevel>0</EncryptionLevel>
<Block>N</Block>
<MasterDb>jdbc:db2:VNURSE</MasterDb>
<MasterDriver>COM.ibm.db2.jdbc.app.DB2Driver</MasterDriver>
<MirrorDb>jdbc:db2:M_VN2</MirrorDb>
<MirrorDriver>COM.ibm.db2.jdbc.app.DB2Driver</MirrorDriver>
<AddTable>
  <TargetSchema>DSYSAMPLE</TargetSchema>
  <TargetTable>VNPERSO</TargetTable>
  <SourceSchema>DSYSAMPLE</SourceSchema>
  <SourceTable>VNPERSO</SourceTable>
  <DeleteAuth>Y</DeleteAuth>
  <InsertAuth>Y</InsertAuth>
  <UpdateAuth>Y</UpdateAuth>
  <SelectAuth>Y</SelectAuth>
  <WhereClauseMirrorToMobile>CITY=:CITY</WhereClauseMirrorToMobile>
  <WhereClauseMasterToMirror></WhereClauseMasterToMirror>
  <JoinFilter>G</JoinFilter>
  <Order>1</Order>
  <Encrypt>N</Encrypt>
  <AddColumn>
    <SourceColumn>ID</SourceColumn>
    <TargetColumn>ID</TargetColumn>
    <Subscribe>Y</Subscribe>
    <Replicate>Y</Replicate>
    <Key>Y</Key>
  </AddColumn>
  <AddColumn>
    <SourceColumn>NAME</SourceColumn>
    <TargetColumn>NAME</TargetColumn>
    <Subscribe>Y</Subscribe>
    <Replicate>Y</Replicate>
    <Key>N</Key>
  </AddColumn>
  <AddColumn>
    <SourceColumn>ADDRESS</SourceColumn>
    <TargetColumn>ADDRESS</TargetColumn>
    <Subscribe>Y</Subscribe>
    <Replicate>Y</Replicate>
    <Key>N</Key>
  </AddColumn>
  <AddColumn>
    <SourceColumn>CITY</SourceColumn>
    <TargetColumn>CITY</TargetColumn>
    <Subscribe>Y</Subscribe>
    <Replicate>Y</Replicate>
    <Key>N</Key>
  </AddColumn>
  <AddColumn>
    <SourceColumn>HOMEPHONE</SourceColumn>
    <TargetColumn>HOMEPHONE</TargetColumn>
    <Subscribe>Y</Subscribe>
    <Replicate>Y</Replicate>
    <Key>N</Key>
  </AddColumn>
  <AddColumn>
    <SourceColumn>WORKPHONE</SourceColumn>
    <TargetColumn>WORKPHONE</TargetColumn>
    <Subscribe>Y</Subscribe>
    <Replicate>Y</Replicate>
    <Key>N</Key>
  </AddColumn>
  <AddColumn>
    <SourceColumn>MOBILEPHONE</SourceColumn>
    <TargetColumn>MOBILEPHONE</TargetColumn>

```

```

        <Subscribe>Y</Subscribe>
        <Replicate>Y</Replicate>
        <Key>N</Key>
    </AddColumn>
    <AddIndex>
        <Name>IDX_CITY</Name>
        <ColNames>+CITY;</ColNames>
    </AddIndex>
</AddTable>
</AddJdbcSubscription>

</Root>

```

Example of removing tables from a jdbc subscription: This example removes two tables from the 'JDBCSUB_2' subscription (assuming subscription had many tables). Tables of a subscription are identified by their target schema and target table names. When tables are excluded from a subscription, these tables in the mirror database will be dropped. This might cause data loss if these tables still have pending synchronization data that is not yet replicated to the source database. Replicate="TRUE" attribute tells the XML tool to replicate existing data before dropping these tables. In some cases, administration might know that there is no data to replicate. Then it is fine to use Replicate="FALSE" to make the script run faster. Also, certain problems that prevent successful replication can be skipped by providing Replicate="FALSE".

```

<?xml version="1.0"?>
<!DOCTYPE Root SYSTEM "file:/com/ibm/mobileservices/xmlscripting/mdas3.dtd">

<Root>
  <AlterJdbcSubscription Replicate="TRUE">
    <Name>JDBCSUB_2</Name>
    <JdbcSubscription>
      <RemoveTable>
        <TargetSchema>DSYSAMPLE</TargetSchema>
        <TargetTable>VNPERSO</TargetTable>
      </RemoveTable>
      <RemoveTable>
        <TargetSchema>DSYSAMPLE</TargetSchema>
        <TargetTable>VNMEDICALRECORD</TargetTable>
      </RemoveTable>
    </JdbcSubscription>
  </AlterJdbcSubscription>
</Root>

```

Adding tables to a JDBC subscription: This example shows how to add tables to an existing subscription. The <Order> tag tells the XML Scripting Tool to make this VNMEDICALRECORD table the 2nd table in this subscription. If there is already more than 1 table, this change will push other tables to 3rd, 4th, 5th place. The devices that are effected by this change will receive new configuration information and will adjust accordingly. If there are less than 2 tables in this subscription, this table will become the last table. Replicate="FALSE" flag tells the XML Scripting tool not to replicate this new table after it is created in the mirror. This will cause the mirror table to be empty and there won't be any data to sync. Therefore, the subscription will remain blocked and devices cannot synchronize until next successful replication that will fill this table with data from source and make it ready for synchronization. While adding a new table to an existing or a new subscription, all columns of that source table must be listed with their synchronization and replication settings.

```

<?xml version="1.0"?>
<!DOCTYPE Root SYSTEM "file:/com/ibm/mobileservices/xmlscripting/mdas3.dtd">
<Root>
  <AlterJdbcSubscription Replicate="FALSE">
    <Name>JDBCSUB_2</Name>
    <JdbcSubscription>
      <AddTable>
        <TargetSchema>DSYSAMPLE</TargetSchema>
        <TargetTable>VNMEDICALRECORD</TargetTable>
        <SourceSchema>DSYSAMPLE</SourceSchema>
      </AddTable>
    </JdbcSubscription>
  </AlterJdbcSubscription>
</Root>

```

```

<SourceTable>VMMEDICALRECORD</SourceTable>
<DeleteAuth>Y</DeleteAuth>
<InsertAuth>Y</InsertAuth>
<UpdateAuth>Y</UpdateAuth>
<SelectAuth>Y</SelectAuth>
<WhereClauseMirrorToMobile></WhereClauseMirrorToMobile>
<WhereClauseMasterToMirror></WhereClauseMasterToMirror>
<JoinFilter>G</JoinFilter>
<Order>2</Order>
<Encrypt>N</Encrypt>
<AddColumn>
  <SourceColumn>RECORDID</SourceColumn>
  <TargetColumn>RECORDID</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>Y</Key>
</AddColumn>
<AddColumn>
  <SourceColumn>DATE_C</SourceColumn>
  <TargetColumn>DATE_C</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>N</Key>
</AddColumn>
<AddColumn>
  <SourceColumn>TIME_C</SourceColumn>
  <TargetColumn>TIME_C</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>N</Key>
</AddColumn>
<AddColumn>
  <SourceColumn>PATIENTID</SourceColumn>
  <TargetColumn>PATIENTID</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>N</Key>
</AddColumn>
<AddColumn>
  <SourceColumn>BLOODPRESSURE</SourceColumn>
  <TargetColumn>BLOODPRESSURE</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>N</Key>
</AddColumn>
<AddColumn>
  <SourceColumn>PULSERATE</SourceColumn>
  <TargetColumn>PULSERATE</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>N</Key>
</AddColumn>
<AddColumn>
  <SourceColumn>TEMPERATURE</SourceColumn>
  <TargetColumn>TEMPERATURE</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>N</Key>
</AddColumn>
<AddColumn>
  <SourceColumn>WEIGHT</SourceColumn>
  <TargetColumn>WEIGHT</TargetColumn>
  <Subscribe>Y</Subscribe>
  <Replicate>Y</Replicate>
  <Key>N</Key>
</AddColumn>
<AddColumn>

```

```

        <SourceColumn>COMMENT</SourceColumn>
        <TargetColumn>COMMENT</TargetColumn>
        <Subscribe>Y</Subscribe>
        <Replicate>Y</Replicate>
        <Key>N</Key>
    </AddColumn>
</AddTable>
</JdbcSubscription>
</AlterJdbcSubscription>
</Root>

```

Changing existing tables of a JDBC Subscription: This example shows how to change the details of existing tables of existing subscriptions: This script modifies the where clauses that do horizontal filtering for replication and synchronization. because Replicate="TRUE" is specified, before any change is made, existing mirror will be replicated to the source database. Then changes will be applied and another replication will be attempted again. This second replication might fail if you have an error in the where clause. You will then need to revert your changes with another <AlterJdbcSubscription> tag with Replicate="FALSE" flag. You can also replicate manually using command line dsyreplicate.bat after your changes are applied.

```

<?xml version="1.0"?>
<!DOCTYPE Root SYSTEM "file:/com/ibm/mobileservices/xmlscripting/mdas3.dtd">
<Root>
  <AlterJdbcSubscription Replicate="TRUE">
    <Name>JDBCSUB_2</Name>
    <JdbcSubscription>
      <AlterTable>
        <TargetSchema>DSYSAMPLE</TargetSchema>
        <TargetTable>VNPERSO</TargetTable>
        <Table>
          <WhereClauseMirrorToMobile>NAME like :PREFIX</WhereClauseMirrorToMobile>
          <WhereClauseMasterToMirror>NAME like 'A%'</WhereClauseMasterToMirror>
        </Table>
      </AlterTable>
    </JdbcSubscription>
  </AlterJdbcSubscription>
</Root>

```

Adding and Removing Indexes: This example adds a new index (IDX1_a) to an existing table and removes an existing index (IDX1_b) from this table. because MirrorOnly="TRUE" is not specified for the index, this index will also be created on target devices which subscribe to this subscription as well.

```

<?xml version="1.0"?>
<!DOCTYPE Root SYSTEM "file:/com/ibm/mobileservices/xmlscripting/mdas3.dtd">
<Root>
  <AlterJdbcSubscription Replicate="FALSE">
    <Name>JDBCSUB_2</Name>
    <JdbcSubscription>
      <AlterTable>
        <TargetSchema>DSYSAMPLE</TargetSchema>
        <TargetTable>VNPERSO</TargetTable>
        <Table>
          <AddIndex>
            <Name>IDX1_a</Name>
            <ColNames>+NAME;</ColNames>
          </AddIndex>
          <RemoveIndex>
            <Name>IDX1_b</Name>
          </RemoveIndex>
        </Table>
      </AlterTable>
    </JdbcSubscription>
  </AlterJdbcSubscription>
</Root>

```

DataPropagator subscriptions: <AddReplMaster> tag defines source database connection information. The system, instance, database and password are required.

<AddReplMirror> tag defines mirror database connection information. The database, user id, password and the sync window (replication frequency) are required. Replication (DPropR) master and mirror information must be defined before a DPropR subscription can reference such databases.

<AddDpropRSubscription> tag defines a DPropR subscription for DB2 Everyplace Sync Server. The replication mechanism between the source and the mirror database must already be setup using Data Propagator tools as if XML Tool or DB2 Everyplace Sync Server never existed. XML Tool picks up from the point where DB2 Everyplace Sync Server specific setup is needed. Refer to Data Propagator documentation to find out how to setup replication sources and a replication subscription to replicate your source and mirror. The mirror tables must already be created before you can use XML Tool to introduce your DPropR replication system to the DB2 Everyplace Sync Server via DB2 Everyplace Sync Server DPropR subscription you are creating here.

<AddDPropRSubscription> tag must contain 1 or more <AddTable> tags. <AddTable> tags should list all the columns of the mirror table to specify replication and synchronization options for each column using <AddColumn> tag. XML Tool gets table descriptions from the mirror database in case of DPropR subscriptions, unlike JDBC subscriptions which use the source database for this. <SourceSchema> and <SourceTable> identify a table in the source database that we are subscribing to.

<TargetSchema> and <TargetTable> identify a table in the mirror database that we are subscribing to. These names will also be used on the mobile devices. <AddTable> tags can specify indexes for the table columns using <AddIndex> tag. Index names need to be unique for the whole system for the sake of simplicity.

Horizontal mirror filters defined with the <WhereClauseMirrorToMobile> tag in DB2 Everyplace 8.1.4 referenced source database column names, which made it hard to support different source databases because of varying degrees of column name compatibility. DB2 Everyplace 8.2 now references mobile database column names instead of source database column names in horizontal mirror filters to better support different source databases. Only new subscriptions use mobile database column names. Migrated subscriptions continue to use source column names in the mirror database. If you are exporting a subscription to XML from DB2 Everyplace 8.1.4, and want to use it to create a new subscription, modify the <WhereClauseMirrorToMobile> tag to reference mobile database target names. Subscriptions that do not have horizontal mirror filters or have the same names for source database and mobile database columns do not need to do this.

The example below creates a DPropR subscription named 'DPSUB1' which uses VNURSE database as the source and M_VNURSE as the mirror. This subscription has only 1 table that has multiple columns and 1 ascending index defined (the '+' before the index column means ascending, '-' means descending). Multiple columns can be specified like "+COL1;-COL2;+COL3" for an index.

```
<?xml version="1.0"?>
<!DOCTYPE Root SYSTEM "file:/com/ibm/mobileservices/xmlscripting
                               /mdas3.dtd">
<Root>
  <AddReplMaster>
    <System>SAD_LPT</System>
    <Instance>DB2</Instance>
    <Database>VNURSE</Database>
    <UserId>db2admin</UserId>
    <Password>db2admin</Password>
    <DatabaseAlias>VNURSE</DatabaseAlias>
  </AddReplMaster>

  <AddReplMirror>
    <Database>M_VNURSE</Database>
    <UserId>db2admin</UserId>
```

```

    <Password>db2admin</Password>
    <SyncWindow>3600</SyncWindow>
</AddReplMirror>

<AddDproprSubscription Replicate="FALSE">
  <Name>DPSUB1</Name>
  <SubAdapter>DSYDPROPR</SubAdapter>
  <EncryptionLevel>0</EncryptionLevel>
  <Block>N</Block>
  <MasterSystem>SAD_LPT</MasterSystem>
  <MasterInstance>DB2</MasterInstance>
  <MasterDb>VNURSE</MasterDb>
  <MirrorDb>M_VNURSE</MirrorDb>
  <AddTable>
    <TargetSchema>DSYSAMPLE</TargetSchema>
    <TargetTable>VNMEDICALRECORD</TargetTable>
    <SourceSchema>DSYSAMPLE</SourceSchema>
    <SourceTable>VNMEDICALRECORD</SourceTable>
    <DeleteAuth>Y</DeleteAuth>
    <InsertAuth>Y</InsertAuth>
    <UpdateAuth>Y</UpdateAuth>
    <SelectAuth>Y</SelectAuth>
    <WhereClauseMirrorToMobile></WhereClauseMirrorToMobile>
    <WhereClauseMasterToMirror></WhereClauseMasterToMirror>
    <JoinFilter>G</JoinFilter>
    <Order>1</Order>
    <Encrypt>N</Encrypt>
    <AddColumn>
      <SourceColumn>RECORDID</SourceColumn>
      <TargetColumn>RECORDID</TargetColumn>
      <Subscribe>Y</Subscribe>
      <Replicate>Y</Replicate>
      <Key>Y</Key>
    </AddColumn>
    <AddColumn>
      <SourceColumn>DATE_C</SourceColumn>
      <TargetColumn>DATE_C</TargetColumn>
      <Subscribe>Y</Subscribe>
      <Replicate>Y</Replicate>
      <Key>N</Key>
    </AddColumn>
    <AddColumn>
      <SourceColumn>TIME_C</SourceColumn>
      <TargetColumn>TIME_C</TargetColumn>
      <Subscribe>Y</Subscribe>
      <Replicate>Y</Replicate>
      <Key>N</Key>
    </AddColumn>
    <AddColumn>
      <SourceColumn>PATIENTID</SourceColumn>
      <TargetColumn>PATIENTID</TargetColumn>
      <Subscribe>Y</Subscribe>
      <Replicate>Y</Replicate>
      <Key>N</Key>
    </AddColumn>
    <AddColumn>
      <SourceColumn>BLOODPRESSURE</SourceColumn>
      <TargetColumn>BLOODPRESSURE</TargetColumn>
      <Subscribe>Y</Subscribe>
      <Replicate>Y</Replicate>
      <Key>N</Key>
    </AddColumn>
    <AddColumn>
      <SourceColumn>PULSERATE</SourceColumn>
      <TargetColumn>PULSERATE</TargetColumn>
      <Subscribe>Y</Subscribe>
      <Replicate>Y</Replicate>
    </AddColumn>
  </AddTable>
</AddDproprSubscription>

```

```

        <Key>N</Key>
    </AddColumn>
    <AddColumn>
        <SourceColumn>TEMPERATURE</SourceColumn>
        <TargetColumn>TEMPERATURE</TargetColumn>
        <Subscribe>Y</Subscribe>
        <Replicate>Y</Replicate>
        <Key>N</Key>
    </AddColumn>
    <AddColumn>
        <SourceColumn>WEIGHT</SourceColumn>
        <TargetColumn>WEIGHT</TargetColumn>
        <Subscribe>Y</Subscribe>
        <Replicate>Y</Replicate>
        <Key>N</Key>
    </AddColumn>
    <AddColumn>
        <SourceColumn>COMMENT</SourceColumn>
        <TargetColumn>COMMENT</TargetColumn>
        <Subscribe>Y</Subscribe>
        <Replicate>Y</Replicate>
        <Key>N</Key>
    </AddColumn>
</AddTable>
</AddDproprSubscription>
</Root>

```

Creating a group or a user:

The <AddGroup> tag defines a Group for DB2 Everyplace Sync Server. Users that can synchronize data with the DB2 Everyplace Sync Server must be associated with existing Groups. The Group is identified by it's name. It can be enabled or disabled for synchronization. A Group can have 1 or more data filters defined using the <AddDataFilter> tag.

These data filters are used in where clauses of subscribed tables to do horizontal filtering. For example, a Group can have a data filter named ':CITY'. The where clause of a table can specify a condition like MYSCHEMA.MYTABLE.CITYCOLUMN=:CITY. Then, in case of this example, if we were to create this Group and synchronize it's users, we would be selecting rows where MYSCHEMA.MYTABLE.CITYCOLUMN='SAN JOSE' by default. Note that, individual users can override a data filter of their Group to specify a different data filter value. A user cannot override a data filter that it's Group does not provide.

In the example below, the User 'nurse_2' is created and assigned to 'GROUP_2'. It overrides data filter :CITY to provide a different city name specific to this 'nurse_2'. However, this user does not override the data filter ':COUNTRY'. It inherits the ':COUNTRY' data filter from it's group 'GROUP_2' with it's value 'USA'.

Lastly, an existing user named 'nurse_3' is assigned to this new group 'GROUP_2'.

```

<?xml version="1.0"?>
<!DOCTYPE Root SYSTEM "file:/com/ibm/mobileservices/xmlscripting/mdas3.dtd">

<Root>
  <AddGroup>
    <Name>GROUP_2</Name>
    <Description>A test group</Description>
    <Enabled>Y</Enabled>
    <AddDataFilter>
      <Name>:CITY</Name>
      <Value>'SAN JOSE'</Value>
    </AddDataFilter>
    <AddDataFilter>
      <Name>:COUNTRY</Name>

```



```

        <Value>'USA'</Value>
    </AddDataFilter>
    <IncludeSubscriptionSet>
<Name>SUBSET_2</Name>
    </IncludeSubscriptionSet>
</AddGroup>

<AddUser>
    <Name>nurse_2</Name>
    <UserGroup>GROUP_2</UserGroup>
    <Description>A test user</Description>
    <Password>nurse_2</Password>
    <OverrideDataFilter>
        <Name>:CITY</Name>
        <Value>SANTA CLARA</Value>
    </OverrideDataFilter>
</AddUser>

<AlterUser>
<Name>nurse_3</Name>
<User>
    <UserGroup>GROUP_2</UserGroup>
</User>
</AlterUser>
</Root>

```

Another example of changing the details of a synchronization user: This script changes the description of a user named nurse_5, sets a new password and overrides a data filter :DF1 that the user inherits from its assigned Group whatever that might be. If the current group of the user does not have a data filter called :DF1, this script will fail and no changes will be applied for this <AlterUser> tag.

```

<?xml version="1.0"?>
<!DOCTYPE Root SYSTEM "file:/com/ibm/mobileservices/xmlscripting/mdas3.dtd">

<Root>
    <AlterUser>
        <Name>nurse_5</Name>
        <User>
            <Description>A new description.</Description>
            <Password>nurse5pass</Password>
            <OverrideDataFilter>
                <Name>:DF1</Name>
                <Value>-1</Value>
            </OverrideDataFilter>
        </User>
    </AlterUser>
</Root>

```

Creating subscription sets by using the XML Scripting tool:

The <AddSubscriptionSet> tag defines a new subscription set. The <IncludeSubscription> tag associates existing subscriptions with a subscription set. In the following example, the subscription set named SUBSET_2 includes existing subscriptions named JDBC SUB_2 and FILE SUB_3. By default, subscriptions are synchronized in the order that they are listed in the XML that defines the subscription set. Therefore, in SUBSET_2, JDBC SUB_2 is synchronized before FILE SUB_3.

```

<?xml version="1.0"?>
<!DOCTYPE Root SYSTEM "file:/com/ibm/mobileservices/xmlscripting
/mdas3.dtd">

<Root>
    <AddSubscriptionSet>
        <Name>SUBSET_2</Name>
        <IncludeSubscription>
            <Name>JDBC SUB_2</Name>
        </IncludeSubscription>
        <IncludeSubscription>

```

```

    <Name>FILESUB_3</Name>
  </IncludeSubscription>
</AddSubscriptionSet>
</Root>

```

You can use the <Order> tag inside the <IncludeSubscription> tag to explicitly specify relative ordering. The following example adds a subscription named JDBC SUB_4 (assuming JDBC SUB_4 already exists) into the subscription set named SUBSET_2. The example uses the <Order> tag to make JDBC SUB_4 the first subscription synchronized.

```

<?xml version="1.0"?>
<!DOCTYPE Root SYSTEM "file:/com/ibm/mobileservices/xmlscripting
                        /mdas3.dtd">
<Root>
  <AlterSubscriptionSet>
    <Name>SUBSET_2</Name>
    <SubscriptionSet>
      <IncludeSubscription>
        <Name>JDBC SUB_4</Name>
        <Order>1</Order>
      </IncludeSubscription>
    </SubscriptionSet>
  </AlterSubscriptionSet>
</Root>

```

XML Scripting tool restrictions

The following restrictions apply to the XML Scripting tool:

- When you create a DataPropagator subscription with the XML Scripting tool, the mirror database must be located on the same machine as the XML Scripting tool if the Replicate attribute is set to TRUE. The same restriction applies when altering or removing a DataPropagator subscription.

Related concepts

“About the XML scripting tool” on page 85

“Overview of the XML scripting tool” on page 85

Related tasks

“Exporting XML scripts” on page 86

Related reference

“Invoking the XML Scripting tool from the command line” on page 86

“Invoking the XML tool programmatically” on page 87

“XML scripting tool DTD”

XML scripting tool DTD

```

<!ELEMENT Root( AddJdbcMaster | AddRepMaster | AddJdbcMirror |
  AddRepMirror | AddGroup | AddUser | AddSubscriptionSet |
  AddJdbcSubscription | AddFileSubscription | AddDproprSubscription |
  AddUploadSubscription | AddCustomSubscription | AddAdapter |
  AddServer | ResetDevice | ResetUser | ResetGroup | ResetAll |
  RemoveDevice | RemoveJdbcMaster | RemoveRepMaster |
  RemoveJdbcMirror | RemoveRepMirror | RemoveGroup | RemoveUser |
  RemoveSubscriptionSet | RemoveSubscription | RemoveAdapter |
  RemoveServer | AlterUser | AlterGroup | AlterDevice |
  AlterFileSubscription | AlterJdbcSubscription |
  AlterUploadSubscription | AlterDproprSubscription |
  AlterSubscriptionSet | AlterJdbcMaster | AlterJdbcMirror |
  AlterRepMaster | AlterRepMirror | AlterAdapter |
  BlockSubscription | UnblockSubscription | AddProperty |
  RemoveProperty | AlterProperty | ReplicateMirror | PruneMirror )*>

```

<!--A NULL element under a node means the node's value is null. This is needed to insert NULL values into the database. Some elements, listed below, allow NULL values to be specified.-->

<!ELEMENT NULL EMPTY>

<!--Master and mirror database related tags:

Sync Server replicates data from the master (source) databases to the mirror databases and then synchronizes the mirror databases with the mobile devices. Therefore a replicated subscription definition must refer to a source and a mirror database.

There are two types of mirror and master databases based on their replication technology. JDBC and DPropR. DPropR master and mirrors are replicated using the DB2 replication. Therefore both the master and mirror must be DB2 databases. JDBC master and mirrors are replicated using Sync Server's JDBC replication system. The master (source) can be any database that has a well behaving JDBC driver. The mirror must still be DB2. Certain data type mappings are done by the Sync Server for this master to mirror replication to succeed if master database is not DB2. Some data types in the source database might not be supported. In case of DPropR subscriptions, the replication setup between the source and the mirror must be performed BEFORE running the XML tool to add a DPropR subscription (AddDpropSubscription tag) to enable synchronization of this mirror to mobile devices. To do this, in the control center, user must define source tables as Replication Source to be able to replicate them. Then the user needs to create a Replication Subscription (which is a DPropR term). DB2e Sync Server expects that Apply Qualifier for a DPropR replication subscription is the same as the mirror database name.

For example,

"M_VNURSE" in our VNURSE sample application. User should enter this mirror database name as Apply Qualifier.

Once this Replication Subscription is created successfully, a user can run the XML Tool to introduce this replication subscription to Sync Server and create the synchronization infrastructure in the mirror and control database.

Replication Subscription name should match

Sync Server DPropR table subscription name

that the user intends to use in the XML tool to avoid complications.

MDAC (DB2e Sync Server admin GUI) manages Replication Subscription creation and Sync Server DPropR subscription creation together without exposing details of the process. Doing so, it uses the same subscription name both for Replication Subscription and the Sync Server DPropR subscription.

-->

<!-- AddJdbcMaster tag inserts connection information for a JDBC master (source) database. This description is referenced from a JDBC subscription related elements to set the subscription's source database. The child elements that this tag contains are as follows:

'Database'(max length=255) is the JDBC URL to the source database.

'Driver'(max length=255) is the JDBC driver class name.

'UserId'(max length=128) is the id used to connect to this database using JDBC.

'Password'(max length=128) is the password used to connect to this database using JDBC.

'ControlTableSpace'(max length=128) is the DB2 or Oracle tablespace to use for Sync Server specific control tables in the source database.-->

<!ELEMENT AddJdbcMaster (Database, Driver, UserId, Password, ControlTableSpace?)>
<!ELEMENT ControlTableSpace (#PCDATA|NULL)*>

<!-- AddRepMaster tag inserts the description of a DPropR master (source) database. This description is referenced from DPropR subscription related elements to set the subscription's source database. The child elements that this tag contains are as follows:

'System'(max length=128) is the DB2 system name for the source database.
(As displayed in DB2 UDB Control Center)

'Instance'(max length=8) is the DB2 instance name for the source database.
(As displayed in DB2 UDB Control Center)

'Database'(max length=18) is the DB2 database name for the source database.
(As displayed in DB2 UDB Control Center)

'UserId'(max length=128) is the id used to connect to this database using JDBC.
 'Password'(max length=128) is the password used to
 connect to this database using JDBC.
 'DatabaseAlias'(max length=18) is the alias used for this database.
 'ControlTableSpace'(max length=128) is the DB2 tablespace to use for
 global control tables in the source database.
 -->

```
<!ELEMENT AddRepMaster (System, Instance, Database,
                        UserId, Password, DatabaseAlias?,
                        ControlTableSpace?)>
  <!ATTLIST AddRepMaster CreatedPropRControlTables (TRUE | FALSE) "FALSE">

  <!ELEMENT System    (#PCDATA)>
  <!ELEMENT Instance  (#PCDATA)>
  <!ELEMENT Database  (#PCDATA)>
  <!ELEMENT UserId    (#PCDATA)>
  <!ELEMENT Password  (#PCDATA|NULL)*><!--Password of databases can't be null,
but password of groups can be null.-->
  <!ATTLIST Password Encryption (NONE|DSY) "NONE">

  <!ELEMENT DatabaseAlias (#PCDATA|NULL)*>
```

<!-- AddJdbcMirror tag inserts the connection information for
 a JDBC mirror database.
 This description is referenced from JDBC subscription related elements to set
 the subscription's mirror database. The child elements that this tag contains
 are as follows:
 'Database'(max length=255) is the DB2 database name for the mirror database.
 (As displayed in DB2 UDB Control Center)
 'Driver'(max length=255) is the JDBC driver class name.
 'UserId'(max length=128) is the id used to connect to this database using JDBC.
 'Password'(max length=128) is the password used to connect
 to this database using JDBC.
 'ControlTableSpace'(max length=128) is the DB2 tablespace to use for
 Sync Server specific global control tables in the mirror database.
 'AddCustomLogic' tag makes it possible to associate one or more user-defined
 CustomLogic implementation classes with this mirror database to customize
 replication behavior. Keep in mind that errors in custom logic implementation
 will cause replication to fail.
 This might block attempted changes to replicated subscriptions if you
 insist on replicating.
 -->

```
<!ELEMENT AddJdbcMirror (Database, Driver, UserId, Password,
                        SyncWindow?, ControlTableSpace?, AddCustomLogic* )>
  <!ELEMENT Driver    (#PCDATA)>
  <!ELEMENT SyncWindow (#PCDATA|NULL)*>
```

<!-- AddRepMirror tag inserts the description of a DPropR mirror database.
 This description is referenced from DPropR subscription related
 elements to set the subscription's mirror database. The child elements that
 this tag contains are as follows:
 'Database'(max length=18) is the DB2 database name for the mirror database.
 (As displayed in DB2 UDB Control Center)
 'UserId'(max length=128) is the id used to connect to this database using JDBC.
 'Password'(max length=128) is the password used to connect
 to this database using JDBC.
 'SyncWindow'(integer) is the time in seconds to repeat replication
 between master and mirror databases.
 'ControlTableSpace'(max length=128) is the DB2 tablespace to use for Sync Server
 and DpropR specific control tables in the mirror database.
 'UOWControlTablespace'(max length=128) is the DB2 tablespace to use for the
 DPropR IBMSNAP_UOW table. Compared to other DpropR control tables, this table
 might grow to be big, thus a separate tablespace option is provided.

'AddCustomLogic' tag makes it possible to associate one or more user-defined CustomLogic implementation classes with this mirror database to customize replication behavior.

```
-->
<!ELEMENT AddReplMirror (Database, UserId, Password,
                        SyncWindow, ControlTableSpace?,
                        UOWControlTablespace?, AddCustomLogic*)>
  <!ATTLIST AddReplMirror CreatedPropRControlTables (TRUE | FALSE) "FALSE">
```

```
<!--Users and groups in Sync Server:
Sync Server uses the concept of a 'user' (see 'AddUser' tag) to associate a
device with subscriptions to be synchronized for that device.
A User belongs to a Group (see 'AddGroup' tag).
A Group contains Subscription Sets (see 'AddSubscriptionSet' tag).
A Subscription Set contains Subscriptions (see 'AddXYZSubscription' tags).
Subscriptions define the subset of the database schema that will be synchronized
with the mobile database.
This XML DTD describes how to create these conceptual entities and associate
them with each other.
-->
```

```
<!--AddGroup tag inserts the description of a group.
'Name'(max length=18) is a unique group name.
'Description'(max length=128) is a text description.
'Enabled' is 'Y' or 'N'. If 'Y', members of this group can synchronize,
if 'N' they cannot.
-->
```

```
<!ELEMENT AddGroup (Name, Description?, Enabled,
                    (IncludeSubscriptionSet | AddDataFilter)*) >
  <!ELEMENT Name (#PCDATA)>
  <!ELEMENT Description (#PCDATA|NULL)*>
  <!ELEMENT Enabled (#PCDATA)>
```

```
<!--IncludeSubscriptionSet tag includes a Subscription Set
(a logical grouping of subscriptions that are synchronized together)
in the Group that contains this tag.
A group must contain non-empty Subscription Sets to describe what kind of data
group members (users) are going to synchronize in order to be able to sync
data.
'Name'(max length=18) is the name of the associated Subscription Set.
'Order' is an integer that tells which Subscription Set comes first. If not
specified, the order they appear inside the 'AddGroup' tag is used.
If the Subscription Set identified by 'Name' does not exist, the AddGroup tag
that wants to include this Subscription Set fails and the group cannot be created.
-->
<!ELEMENT IncludeSubscriptionSet (Name, Order?)>
```

```
<!--AddDataFilter tag declares a data filter for the Group that
contains this tag. Data filters make it possible to send a portion of a source
table to the mobile database given a group or a user. They also help create the
target (mobile) database slightly differently based on the user's data filters.
(Please see 'ColumnDataFilter' and 'Constnt' tags). A data filter consists of
a 'Name' and a 'Value'.
```

'Name'(max length=18) specifies any name that will be referenced in the actual filtering where clause (Such where clauses are specified per table in Subscription settings).

'Value'(max length=36) specifies the default value to replace the parameter name. (each user's settings can override this).

The data to be synchronized can be filtered by a condition like: 'ACOLUMN IN (SELECT ACOLUMN FROM TABLE :name=value)' in the mirror-to-target where clause of a Subscription Table.

For an 'AddUser' tag to have an 'OverrideDataFilter' tag, the user's group MUST have the same data filter created. In other words, data filters are created for groups, and sometimes overridden by users.

The reason data filters are created at the group level is that, groups contain Subscriptions that use these data filters in their row filtering where clauses. If some users did not have a given data filter, they would not be able to sync a subscription that used the data filter, therefore, the relationships are such that a group that has a subscription which has (mirror to mobile) row filtering, must have the related data filters defined, and users inherit these data filters. Defining more data filters that are not used by the subscriptions do not have an effect and are ignored during synchronization.

Data filters cannot be used in master to mirror where clause filters to filter replicated rows. The reason of this is that replication does not run in the context of a group, it is independent of all groups, therefore a group's data filter is irrelevant and if used by mistake, will likely cause SQL errors during replication.

(See also 'ColumnDataFilter' and 'Constrnt' tags).

-->

```
<!ELEMENT AddDataFilter (Name, Value?) >
  <!ELEMENT Value (#PCDATA|NULL)*>
```

```
<!ELEMENT OverrideDataFilter (Name, Value?) >
```

<!--AddUser tag creates a user. Every mobile database must have a corresponding user to sync with Sync Server.

'Name'(max length=18) is the unique name of a user.

'UserGroup'(max length=18) is the name of an existing group. If such group does not exist, AddUser fails and user cannot be added.

'Description'(max length=128) is text description of the user.

'Password'(max length=18) is the password that user must specify on the mobile device in order to synchronize with Sync Server.

'OverrideDataFilter' element allows this user to provide a custom value for a data filter added to the user's group. Please see comments for 'AddDataFilter' tag.

-->

```
<!ELEMENT AddUser (Name, UserGroup?, Description?, Password?, LDAPProperties?,
  (OverrideDataFilter | AddDevice)*) >
  <!ELEMENT UserGroup (#PCDATA|NULL)*>
```

<!--AddDevice tag adds a device description for the user. Normally, such descriptions are created by the Sync Server during synchronization based on the actual device that is performing the sync. However, this tag is provided to accomodate certain test cases and backups that might need this functionality.

'Id'(max length=32) is generated by the server.

'Type' is a string that refers to an enumerated device type. The string constants we use for this purpose are listed below. The device type used in this tag is one of them. These device types map to integer numbers in the control database. Users can use these numbers in the XML as well (in case there is a new device type in the last minute and XML tool does not know about it).

Possible device type strings are:

PALM or PALM_M68K

```

WINCE          or WINCE_ARM
WIN32          or WIN32_X86
WIN32UNICODE  or WIN32UNICODE_X86
NEUTRINO      or NEUTRINO_X86
  LINUX        or LINUX_X86
EMBEDDEDLINUX or EMBEDDEDLINUX_X86
EPOC          or EPOC_ARM
EPOCUNICODE   or EPOCUNICODE_ARM
DB2J

```

We have also defined a set of other possible variants based on device processor, however these are not used by the Sync Server yet.

So if, for example, XML specifies WIN32_MIPS as device type, then it will default to WIN32 because we do not yet differentiate between processors. Possible string constants are:

```

WINCE_SH3
WINCE_SH4
WINCE_MIPS
WINCE_X86
EPOC_X86
EPOCUNICODE_X86

```

DB2J is for Cloudscape clients.

'Enabled' field is one character, either 'Y' or 'N'.

```
-->
```

```
<!ELEMENT AddDevice (Id, Type, Enabled)>
  <!ELEMENT Id (#PCDATA) >
```

```
<!--
```

In case of LDAP enabled setup, user and group information for the Sync Server are retrieved from the LDAP server and stored in the control database to be associated with subscription sets. While creating a user using XML Tool, LDAPProperties of the user needs to be specified to be used to create the user on the LDAP server. FirstName and LastName are required. To be able to synchronize using Sync Server, an LDAP user must also belong to a 'Sync Group'. To associate a user with a sync group, one or more AddToSyncGroup tags can be used. Sync groups are defined in LDAP specific properties of Sync Server installation. A sync group name should start with the characters 'DB2e'.

Note: During migration of non-ldap setup to ldap enabled setup, Sync Server will use 'DSY' as default FirstName and LastName to create users on LDAP server.

```
-->
```

```
<!ELEMENT LDAPProperties (FirstName, LastName, AddToSyncGroup*,
  RemoveFromSyncGroup*, PreferredLanguage?)>
  <!ELEMENT FirstName (#PCDATA)>
  <!ELEMENT LastName (#PCDATA)>
  <!ELEMENT AddToSyncGroup (#PCDATA)>
  <!ELEMENT RemoveFromSyncGroup (#PCDATA)>
  <!ELEMENT PreferredLanguage (#PCDATA)>
```

<!--A subscription set is a logical group of subscriptions that are synchronized together. A group of users synchronizes against a defined subscription set.

Not individual subscriptions.

'Name'(max length=18) is the name of the Subscription Set.

'Description'(max length=128) is a text description.

'Signature'(max length=128) - not used.

'IncludeSubscription' references an existing subscription that should be included in this subscription set. If the subscription does not exist, creation of this subscription set fails.

```
-->
```

```
<!ELEMENT AddSubscriptionSet (Name, Description?, Signature?,
  (IncludeSubscription*))>
  <!ELEMENT Signature (#PCDATA|NULL)*>
```

```

<!--IncludeSubscription tag is used by AddSubscriptionSet tags.
'Name'(max length=18) is the name of the existing subscription being included.
'Order' is an integer identifying the order of the subscription among other
subscriptions that are also included in the parent Subscription Set.
Subscriptions included in a Subscription Set are synchronized in this order.
If not specified, order of listing in the XML will be used.
-->
<!ELEMENT IncludeSubscription (Name, Order?)>

<!--AddJdbcSubscription tag creates a JDBC Subscription to synchronize relational
data from a mirror database to the mobile databases. Replication from master to
mirror will also be handled by the Sync Server and not DPropR.
'Name'(max length=18) is the unique name of the subscription.
'Description'(max length=128) is a text description.
'SubAdapter'(max length=18) is the name of an existing adapter that
this subscription uses.
'EncryptionLevel' is an integer identifying the encryption level to use.
0 indicates no encryption. 1 indicates 56 bit, and 2 indicates 128 bit encryption.
'Block' is either 'Y' or 'N'. When set to 'Y', the subscription cannot be
synchronized. Target device will be notified that subscription is not available.

'MasterDb'(max length=255) is the master database JDBC url.
'MasterDb' together with 'MasterDriver' references an already existing
master database definition that is shared by all subscriptions that use it.

'MasterDriver'(max length=255) is the JDBC driver to
connect to the master database.
The database definition referenced by the composite key '
MasterDb' and 'MasterDriver' must already be defined.
(Please see AddJdbcMaster tag for more info).

'MirrorDb' and 'MirrorDriver' tags refer to an already existing mirror database
definition just like MasterDb does for a master database.

'AddTable' tag adds database tables in this subscription to be
synchronized and/or replicated.

Note that, upload subscriptions (which do not have mirrors, and only inserts
from mobile to source database) are handled by 'AddUploadSubscription' tag.

Warning: Adding a subscription blocks effected parts of the system (effected mirror
database is a major component) during this operation and makes it unusable
by the server. Avoid such frequent operations while the servers are running or
this will degrade server performance.

'Replicate' attribute is either "TRUE" or "FALSE". It must be specified.
After creating a subscription, a replication is performed to fill the new mirror
tables with data to be ready to synchronize. If replication is skipped by providing
Replicate="FALSE", then the subscription will be blocked (unavailable for sync)
until replication is performed either manually or as part of scheduled replication
for the related mirror. When adding, dropping, editing multiple subscriptions,
for performance, it might be desirable to set Replicate="FALSE" to avoid initial
and final replication attempts and manually perform replication before and after
changes. In some cases, errors in subscription setup can also prevent any further
replication. In that case, replication has to be skipped in order to be able to
edit the subscription to apply fixes. To replicate manually, users can use
"dysyreplicate" command line tool.
-->
<!ELEMENT AddJdbcSubscription (Name, Description?, SubAdapter,
                               EncryptionLevel?, Block?, MasterDb, MasterDriver,
                               MirrorDb, MirrorDriver, (AddTable*))>

  <!ATTLIST AddJdbcSubscription Replicate (TRUE | FALSE) "TRUE">

  <!ELEMENT SubAdapter (#PCDATA)>
  <!ELEMENT EncryptionLevel (#PCDATA)>

```



```

<!ELEMENT Block (#PCDATA)>
<!ELEMENT MasterDb (#PCDATA)>
<!ELEMENT MasterDriver (#PCDATA)>
<!ELEMENT MirrorDb (#PCDATA|NULL)*>
<!ELEMENT MirrorDriver (#PCDATA)>

```

<!--AddUploadSubscription creates a subscription that allows inserts from a mobile database into the source database. Uploads subscriptions do not have associated mirror databases. Please see AddJdbcSubscription tag comments for descriptions of child fields. 'AddTable' tag adds database tables in this subscription to be synchronized. -->

```

<!ELEMENT AddUploadSubscription (Name, Description?, SubAdapter,
                                EncryptionLevel?, Block?, MasterDb, MasterDriver,
                                (AddTable*))>

```

<!--AddDpropRSubscription creates a Sync Server DPropR subscription to synchronize relational data from a mirror database to the mobile databases. Replication from master to mirror will be handled by the DB2 DPropR replication. DB2 DPropR has the concept of a Replication Subscription. This is different than our Sync Server subscriptions we are describing here. The difference between JDBC subscriptions and the DPropR subscriptions is the type of replication employed. because DPropR replication works between two DB2 databases, the master(source) database must be DB2, unlike JDBC subscription where it can be any JDBC compliant database.

The fields Name, Description, SubAdapter, EncryptionLevel and Block are same as other subscriptions. The DPropR Subscription specific fields are as follows:
'MasterSystem'(max length=128) is the DB2 master database instance.
'MasterInstance'(max length=8) is the DB2 master database instance.
'MasterDb'(max length=18) is the master database name. MasterSystem, MasterInstance and MasterDb alltogether form a composite key that refers to an existing master database description, typically created by 'AddReplMaster' tag. If such database definition does not exist, creation of this subscription will fail.
'MirrorDb'(max length=18) is the mirror database name that references a mirror database definition that is typically created by AddReplMirror tag. If such description does not exist, creation of this subscription fails. Sync Server assumes that Apply Qualifier of a replication subscription is the same as this 'MirrorDb' name.

'CommitCount' is the number of rows that indicates how often the transactions of apply process should be committed by DpropR apply process. The default is 0, which means transaction will be committed after all changes are complete.

Warning: Adding a subscription blocks effected parts of the system (effected mirror database is a major component) during this operation and makes it unusable by the server. Avoid such frequent operations while the servers are running or this will degrade server performance.

```

-->
<!ELEMENT AddDpropRSubscription (Name, Description?, SubAdapter,
                                EncryptionLevel?, Block?, MasterSystem,
                                MasterInstance, MasterDb, MirrorDb,
                                ApplyQualifier?, CommitCount?, (AddTable*))>

```

```

<!ATTLIST AddDpropRSubscription Replicate (TRUE | FALSE) "TRUE">
<!ATTLIST AddDpropRSubscription SetupDPropRSourceAndMirror (TRUE|FALSE)"FALSE">

<!ELEMENT MasterSystem (#PCDATA)>
<!ELEMENT MasterInstance (#PCDATA)>
<!ELEMENT ApplyQualifier (#PCDATA)>
<!ELEMENT CommitCount (#PCDATA)>

```

<!--AddFileSubscription creates a subscription to synchronize files from source system to a mobile device. Specific child fields are:
'DeviceTypes' is a semicolon (;) seperated list of device types that this

file can be sent to. This is needed because certain file formats need be different for different devices. Possible device types are listed in the comments for 'AddDevice' tag. 'SourceFileName'(max length=255) is the name of the file in the source system. 'TargetFileName'(max length=255) is the name of the file in the target system.

```
-->
<!ELEMENT AddFileSubscription (Name, Description?, SubAdapter, EncryptionLevel?,
    Block?, DeviceTypes?,SourceFileName, TargetFileName?)>
  <!ELEMENT SourceFileName (#PCDATA)>
  <!ELEMENT TargetFileName (#PCDATA|NULL)*>
  <!ELEMENT DeviceTypes (#PCDATA|NULL)*>
```

<!--AddCustomSubscription creates a custom subscription. The only specific field is 'Other' (max length=4096) which holds application specific data.

```
-->
<!ELEMENT AddCustomSubscription (Name, Description?, SubAdapter?,
    EncryptionLevel?, Block?, Other?)>
  <!ELEMENT Other (#PCDATA|CDATA|NULL)*>
  <!ATTLIST Other Encoding (NONE|Base64) "NONE">

  <!ELEMENT Type (#PCDATA)>
```

<!--AddTable tag provides ability to include tables in the JDBC, DPropR, and upload subscriptions. 'TargetSchema'(max length=128) is the schema of the table on the target device. 'TargetTable'(max length=128) is the name of the table on the target device. TargetSchema and TargetTable together must be unique for a given subscription. In case of DPropR subscriptions, TargetSchema.TargetTable must exist in the mirror database or the parent subscription tag will fail.

'SourceSchema'(max length=128) is the schema of the table in the source database. 'SourceTable'(max length=128) is the name of the table in the source database. The table specified by SourceSchema.SourceTable must exist in the source database in case of JDBC and Upload subscriptions or the creation of the parent subscription tag will fail.

'DeleteAuth' is either 'Y' or 'N', default is 'Y'.
'InsertAuth' is either 'Y' or 'N', default is 'Y'.
'UpdateAuth' is either 'Y' or 'N', default is 'Y'.
'SelectAuth' is either 'Y' or 'N', default is 'Y'.
Changing privileges does not cause the table to be dropped and re-created on the mirror and the target devices.

'WhereClauseMirrorToMobile' (max length=1024) is a constraint part of a DB2 where clause sql statement that will be executed on the mirror to filter rows to be synchronized. It must evaluate to either true or false. It can include data filter parameters defined for groups (see AddDataFilter tag). If data filters are defined, they will be replaced by the values specific to the user or the group of the user being synchronized. Changing this where clause causes the table to be dropped and recreated on the target devices and the mirror database.

For example, lets assume an application has two groups of users, one east coast, one west coast group. Then the where clause for a table might be something like:

```
AREA=:GROUP_AREA
```

Then the group data filters can be defined such that for the west coast group data filter would be:

```
:GROUP_AREA='WESTCOAST'
```

For a west coast group, at run time the where clause will become:

AREA='WESTCOAST'

and the users in the west coast group will get only those rows that match this constraint.

'WhereClauseMasterToMirror'(max length=1024) is constraint portion of a where clause that will be used to filter rows that are replicated from master to the mirror database. because replication does not happen in the context of a user or a group, data filters cannot be used in WhereClauseMasterToMirror. Changing this where clause property of a table causes the table to be dropped and recreated on the mirror and on the target devices.

'JoinFilter' is one of 'Y', 'N' or 'G' for Global (default setting). It specifies whether join filtering should be employed for this table or not. 'Y' means enabled. If the row-filter out-of-scope problem is acceptable, this should be 'N' (disabled) to improve performance. If this tag is not specified, then default value of 'G' will be used, which means that whether join filtering is enabled depends on the global setting RowFilter.OutOfScope.Delete property (located in the Sync Server control database, DSY.PROPERTIES table). If it's value is 1, it is enabled, if it is 0, it is disabled.

'TableSpace' is optional field for the DB2 tablespace to be used in the mirror database where this source table will be replicated to. Sometimes, the default tablespace page size in the mirror database is not big enough to hold replicated data, in this case it is necessary to create the related mirror table in a different table space. Changing the Tablespace of a table will cause the table to be dropped and recreated on the target device and on the mirror.

'SourceTableSpace' is optional field that identifies the tablespace to use in the DB2 source database for the Sync Server specific control tables about this source table in the source database. This 'SourceTableSpace' should not be used if source database is not DB2.

'Order' is an integer that identifies the order of this table inside the subscription. If not specified, order of listing is used. Index of first table is 1. Second is 2 and so on. When used inside 'AlterTable' to change the order of a table, you need to specify the new ordering index for this table. For example, if you want your table to be the 3rd table within the other tables of the subscription, you specify order 3. If the order number you specify is greater than the number of tables, then your table will become the last table in the subscription (assuming there is no other table that asked to be located at even greater index). If the order is 0, or negative, then it will become the first table. You can specify mixed ordering for tables. For example, you can list table A at order 3, and then you can specify table B at order 1, and then table C at order 2. This is to make it easier to change the order of tables in existing scripts without having to move around blocks of AddTable related text. However, having mixed orderings makes the XML Script hard to read and this is discouraged. It is best to use the order that the tables appear when creating subscriptions.

When the Order of a table is changed, the target devices that have not yet heard about this change will receive full description of the full set of tables and they will do re-ordering as needed. This will not cause mirror or target device tables to be dropped and recreated.

'Encrypt' is either 'Y' or 'N'. Default is 'N'. It specifies if table data should be encrypted on the device or not. This is different than the Encryption specified at Subscription level which specifies the encryption level used over the wire during sync. Changing this attribute will cause the table to be dropped and recreated on the target device and on the mirror.

'AddColumn' includes columns in a particular table (Please see comments for AddColumn tag). Adding/removing/altering columns will cause the table to be dropped and recreated on the target device and on the mirror.

'AddIndex' includes indexes in a particular table (Please see comments for AddIndex tag). Changes to the set of indices of a table does NOT cause the table to be dropped and recreated on the target device and on the mirror.

'AddCustomLogic' associates a user defined Java class to provide additional business logic functionality related to this table in Sync Server for synchronization and/or replication purposes. Replication is sometimes a pre-requisite of editing a subscription. Changing the CustomLogic definitions for a table will NOT cause the table to be dropped and recreated on the target device and on the mirror. -->

```
<!ELEMENT AddTable (TargetSchema?, TargetTable?, SourceSchema, SourceTable,
    DeleteAuth?, InsertAuth?, UpdateAuth?, SelectAuth?,
    WhereClauseMirrorToMobile?, WhereClauseMasterToMirror?,
    JoinFilter?, TableSpace?, SourceTableSpace?, Order?,
    Encrypt?, (AddColumn | AddIndex | AddCustomLogic)*, Info?)>
<!ELEMENT TargetSchema (#PCDATA)>
<!ELEMENT TargetTable (#PCDATA)>
<!ELEMENT SourceSchema (#PCDATA)>
<!ELEMENT SourceTable (#PCDATA)>
<!ELEMENT DeleteAuth (#PCDATA)>
<!ELEMENT InsertAuth (#PCDATA)>
<!ELEMENT UpdateAuth (#PCDATA)>
<!ELEMENT SelectAuth (#PCDATA)>
<!ELEMENT WhereClauseMirrorToMobile (#PCDATA|NULL)*>
<!ELEMENT WhereClauseMasterToMirror (#PCDATA|NULL)*>
<!ELEMENT Order (#PCDATA|NULL)*>
<!ELEMENT Encrypt (#PCDATA)>
<!ELEMENT TableSpace (#PCDATA|NULL)*>
<!ELEMENT JoinFilter (#PCDATA)>
<!ELEMENT SourceTableSpace (#PCDATA|NULL)*>

<!ELEMENT Info (MirrTableInfo|SrcTableInfo)*>
<!ELEMENT MirrTableInfo (Schema,TableName)>
<!ELEMENT SrcTableInfo (Schema,TableName)>
<!ELEMENT Schema (#PCDATA|NULL)*>
<!ELEMENT TableName (#PCDATA)>
```

<!--AddColumn tags are used to include columns to a table to be synchronized or replicated.

'SourceColumn' (max length=512) is the column name in the source table. For JDBC and upload subscriptions, column with this name must exist in the source table, otherwise creation of the parent subscription will fail. For DPropR subscriptions it must exist in the mirror database. While running the XML tool, such source and mirror databases must be accessible.

'TargetColumn' will be the name of the column on the mobile device. This name must be unique among the columns of the parent table. If omitted, source name will be used instead.

'Subscribe' is either 'Y' or 'N'. Default is 'Y'. When 'Y', it means column will be synchronized. This 'Subscribe' element is optional.

'Replicate' is either 'Y' or 'N'. Default is 'Y'. When 'Y', it means column will be replicated between master and mirror databases. You cannot synchronize a column without replicating it. This 'Replicate' element is optional.

'Key' is either 'Y' or 'N'. Default is the state in the source database. It indicates if this column is part of a unique key constraint. Some source databases do not have primary key concepts and this value needs to be set in the XML script in that case such that for Sync Server replication and sync purposes, we will treat this column as if it is part of a primary key. For databases that do define the primary keys of their tables, this Key element

need not be specified in the script. When specified, it must match source database setting.

'Constrnt' is a DB2 Everyplace constraint which gets used on the target (mobile) database while creating the column using SQL DDL. For example, to have a constraint on the related column as:

```
constraint CK_ENABLED check(ENABLED in('N','Y'))
```

the user will have to specify <Constrnt>in('N','Y')</Constrnt> in XML.

'ColumnDataFilter' is used to provide default value in SQL DDL while creating a column on a target (mobile) database. Because such a database belongs to a particular 'device user', this data filter can contain Data Filters created for groups and overridden by users (Please see AddDataFilter and OverrideDataFilter tags).

For example, to create a column such that its default value depends on that specific device user's location, SQL DDL could be something like:

```
CITY varchar(50) not null with default 'ISTANBUL'
```

and to make this work polymorphically for each user who has different locations, the administrator will have to specify something like

```
<ColumnDataFilter>:CITY</ColumnDataFilter>
```

as the data filter for this column. The data filter will have to be valid (created) for all groups and users who synchronize this subscription which this column belongs to.

The ColumnDataFilter must be specified for columns that are replicated but not subscribed and not nullable. Because such a column is not synchronized, data will not be coming from the device. The synchronization user specific data filter's runtime value will be used for such columns in place of the data that would be received from a device if the column were subscribed.

-->

```
<!ELEMENT AddColumn (SourceColumn, TargetColumn?, Subscribe?,
                    Replicate?, Key?, Constrnt?, ColumnDataFilter?) >
  <!ELEMENT SourceColumn (#PCDATA)>
  <!ELEMENT TargetColumn (#PCDATA)>
  <!ELEMENT Subscribe (#PCDATA)>
  <!ELEMENT Replicate (#PCDATA)>
  <!ELEMENT Key (#PCDATA)>
  <!ELEMENT Constrnt (#PCDATA|NULL)*>
  <!ELEMENT ColumnDataFilter (#PCDATA|NULL)*>
```

<!--'AddIndex' tag creates index descriptions for target tables.

Index name must be unique within the whole system regardless of subscription or mirror. If MirrorOnly attribute is set to "TRUE", then the index will only be created in the mirror database. It will not be created on the target device. MirrorOnly attribute defaults to "FALSE" if not specified.

'ColNames' tag lists the columns of this index. Column names are the target names used in the AddTable tag. The names are ended with semi colons (;). A plus (+) that precedes a column name means ascending index, a minus (-) means descending index.

AddIndex tag looks like this:

```
<AddIndex>
  <Name>MYINDEX</Name>
  <ColNames>+MYCOLNAME1;+MYCOLNAME2;</ColNames>
</AddIndex>
```

-->

```
<!ELEMENT AddIndex (Name, ColNames)>
<!ATTLIST AddIndex MirrorOnly (TRUE | FALSE) "FALSE">
```

```
<!ELEMENT ColNames (#PCDATA)>
```

<!--

AddCustomLogic tag is used to associate a user defined class with a subscription table or a mirror database to provide custom business logic capability in the synchronization and/or replication process.

'PolicyType' is a constant one of (without the quotes):

```
"BeforeTableSyncInPolicy"  
"AfterTableSyncInPolicy"  
"BeforeTableReplicatePolicy"  
"AfterTableReplicatePolicy"  
"BeforeDbReplicatePolicy"  
"AfterDbReplicatePolicy"  
"TableReplicatePolicy"  
"TableSyncInPolicy"
```

'ClassName' is the fully qualified name of the Java class that implements the custom logic. This class must be located on the CLASSPATH of the Sync Server and the admin tools.

'Enabled' is either 'Y' or 'N'.

-->

```
<!ELEMENT AddCustomLogic (PolicyType, ClassName, Enabled)>  
  <!ELEMENT PolicyType (#PCDATA)>
```

```
<!ELEMENT AddAdapter (Name, Description?, FileAttributes?,  
  CommAttributes?, Signature?, ClassName?)>  
  <!ELEMENT FileAttributes (#PCDATA|NULL)*>  
  <!ELEMENT CommAttributes (#PCDATA|NULL)*>  
  <!ELEMENT ClassName (#PCDATA|NULL)*>
```

<!--AddServer tag creates new a server description. Servers must be defined for the system to work correctly. When a Sync Server starts, it updates the record about itself. If a server goes down without being able to update the record as 'Offline', it might cause problems while editing subscriptions. The administration tool might refuse to apply changes thinking that it is not able to communicate with a server that is up and running in which case making changes would cause unknown behavior. Before making any changes, administrative tools block related parts of the system and communicate the updates to the running servers that are not marked as 'Offline'. If a server is not marked offline and does not respond, it is not safe to apply changes and such changes will be refused.

-->

```
<!ELEMENT AddServer (Name, HostIP, Port, Description?, SyncEnabled?,  
  ReplEnabled?, Status?, Blocked?)>  
  <!ELEMENT HostIP (#PCDATA)>  
  <!ELEMENT Port (#PCDATA)>  
  <!ELEMENT SyncEnabled (#PCDATA)>  
  <!ELEMENT ReplEnabled (#PCDATA)>  
  <!ELEMENT Blocked (#PCDATA)>  
  <!ELEMENT Status (#PCDATA)>
```

<!--'ResetUser' causes the device information of a user to be deleted.

'Name' is the user name.

Resetting a user causes the target device to fo a full refresh next time it tries to synchronize.

-->

```
<!ELEMENT ResetUser (Name)>
```

```
<!ELEMENT ResetDevice (Id)>
```

<!--'ResetGroup' causes the device information of all users in a group to be deleted. 'Name' is the group name.

-->

```
<!ELEMENT ResetGroup (Name)>
```

```

<!--'ResetAll' causes the device information of all users in the system
to be deleted.
-->
<!ELEMENT ResetAll EMPTY>

<!--Removes are below this point-->

<!--
'RemoveJdbcMaster' removes records of a JDBC master database system
that might be shared between subscriptions. It is usually not necessary
to remove these records. It might be useful for some test automation
cases. If a master or mirror is referenced from a subscription, it
cannot be removed and an error will be issued.
-->
<!ELEMENT RemoveJdbcMaster (Database, Driver)>

<!--
'RemoveReplMaster' removes records of a DPropR master database system
that might be shared between subscriptions. If a master or mirror is
referenced from a subscription, it cannot be removed and an error will
be issued.
-->
<!ELEMENT RemoveReplMaster (System, Instance, Database)>
<!ATTLIST RemoveReplMaster DropDPropRControlTables (TRUE | FALSE) "FALSE">

<!--
'RemoveJdbcMirror' removes records of a JDBC mirror database system
that might be shared between subscriptions. If a master or mirror is
referenced from a subscription, it cannot be removed and an error will
be issued.
-->
<!ELEMENT RemoveJdbcMirror (Database, Driver)>

<!--
'RemoveReplMirror' removes records of a DPropR mirror database system
that might be shared between subscriptions. If a master or mirror is
referenced from a subscription, it cannot be removed and an error will
be issued.
-->
<!ELEMENT RemoveReplMirror (Database)>
<!ATTLIST RemoveReplMirror DropDPropRControlTables (TRUE | FALSE) "FALSE">

<!--'RemoveGroup' removes a group and all related data from the system.
The users that used to be in this group will no longer belong to any group.
-->
<!ELEMENT RemoveGroup (Name)>

<!--'RemoveUser' removes a user and all related device data from the system.
Once removed, a user can no longer synchronize.
-->
<!ELEMENT RemoveUser (Name)>

<!--'RemoveServer' removes a server record from the system. For example,
this might be useful when a server name is changed and old name is no
longer valid.
-->
<!ELEMENT RemoveServer (Name)>

<!--'RemoveDevice' is used to remove a device by specifying its device
id using 'Id' tag, or to remove all devices of a user by specifying the

```

user name with 'UserName' tag. The device information will be deleted from the control database. Next time the device attempts to sync again, it's data will be refreshed and the changes on the mobile device will be lost. Use 'ResetDevice' to request a refresh while minimizing data loss.

-->

```
<!ELEMENT RemoveDevice (Id|UserName)>
  <!ELEMENT UserName (#PCDATA)>
```

<!--'RemoveSubscriptionSet' removes a Subscription Set from the system. Notifies online servers about the change. If a server is believed to be online but is not responding, change cannot be applied.

-->

```
<!ELEMENT RemoveSubscriptionSet (Name)>
```

<!--'RemoveSubscription' removes a Subscription from the system. Notifies online servers about the change. If a server is believed to be online but is not responding, change cannot be done. Removing a subscription blocks effected parts of the system (effected mirror database is a major component) during this operation and makes it unusable by the server. Avoid such frequent operations while the servers are running or this will degrade server performance.

-->

```
<!ELEMENT RemoveSubscription (Name)>
```

<!--Replicate attribute is meaningful only for replicated subscriptions such as jdbc.-->

```
<!ATTLIST RemoveSubscription Replicate (TRUE | FALSE) "TRUE">
```

<!--When StopOnError is FALSE, even if source and/or mirror is not accesible, or there is a problem that prevents replication and/or cleanup of source or mirror, subscription will still be deleted from control database. Source and/or mirror might need manual cleanup after that. This is useful in test/development environments where developer does not care about source and mirror databases or they might already be dropped.-->

```
<!ATTLIST RemoveSubscription StopOnError (TRUE | FALSE) "TRUE">
```

```
<!ATTLIST RemoveSubscription DropDPropRSourceAndMirror (TRUE | FALSE) "FALSE">
```

<!--'RemoveAdapter' removes an adapter record from the database and is useful only for some test cases and is not needed most of the time.

-->

```
<!ELEMENT RemoveAdapter (Name)>
```

<!--Blocks a subscription and then notifies online servers about this. Blocked subscriptions cannot sync.-->

```
<!ELEMENT BlockSubscription (Name)>
```

```
<!ELEMENT UnblockSubscription (Name)>
```

<!-- Alters are below this point. -->

<!--

AlterXYZ group of tags are used to locate information and change portions of it. The <AlterXYZ> tags contain the unique identifier for a record to change and the list of properties to change for this record.

For example, the structure of alter user tag looks like this:

```
<AlterUser>
```

```
  <Name>username</Name> //describes which user to change.
```

```
  <User>
```

```
    //lists new values to apply here.
```

```
  </User>
```



```

<AlterUser>
-->

<!--'AlterJdbcMaster' tag allows changing connection information for a
jdbc master database. Most of the time Password is the only property
that might need to be changed. Avoid changing UserId if there is an
existing jdbc subscription using this master as this UserId is used
as schema name in some cases. ControlTableSpace cannot be changed if
jdbc master is already being used by a subscription.
-->
<ELEMENT AlterJdbcMaster (Database, Driver, JdbcMaster)>
<ELEMENT JdbcMaster (Database?, Driver?, UserId?, Password?,
ControlTableSpace?)>

<!--'AlterJdbcMirror' is used to change connection properties of a
jdbc mirror database. It is possible to add/remove one or more custom
logic implementation classes to the mirror. To change existing custom
logic descriptions, remove old description and add new description.
-->
<ELEMENT AlterJdbcMirror (Database, Driver, JdbcMirror)>
<ELEMENT JdbcMirror (Database?, Driver?, UserId?, Password?,
SyncWindow?, (AddCustomLogic | RemoveCustomLogic)*)>

<!--'AlterReplMaster' allows editing of existing DPropR master
database information.
-->
<ELEMENT AlterReplMaster (System, Instance, Database, ReplMaster?)>
<!ATTLIST AlterReplMaster CreateDPropRControlTables (TRUE | FALSE) "FALSE">
<!ATTLIST AlterReplMaster DropDPropRControlTables (TRUE | FALSE) "FALSE">

<ELEMENT ReplMaster (System?, Instance?, Database?, UserId?,
Password?, DatabaseAlias?)>

<!--'AlterReplMirror' allows editing of existing DPropR mirror database
information. It is possible to add/remove one or more custom logic
implementation classes to the mirror. To change existing custom logic
descriptions, remove old description and add new description.
-->
<ELEMENT AlterReplMirror (Database, ReplMirror?)>
<!ATTLIST AlterReplMirror CreateDPropRControlTables (TRUE | FALSE) "FALSE">
<!ATTLIST AlterReplMirror DropDPropRControlTables (TRUE | FALSE) "FALSE">

<ELEMENT ReplMirror (Database?, UserId?, Password?, SyncWindow?,
(AddCustomLogic | RemoveCustomLogic)*)>

<!--'AlterAdapter' can be used to change the description of an adapter. This
is nearly never required.
-->
<ELEMENT AlterAdapter (Name, Adapter)>
<ELEMENT Adapter (Name?, Description?, FileAttributes?,
CommAttributes?, Signature?, ClassName?)>

<!--'AlterFileSubscription' tag can be used to list new settings for a
file subscription. Synchronization requests to the online servers will be blocked
during changes for a short period of time.
'Reset' tag which is an EMPTY tag, instructs the XML Scripting tool to mark
the file subscription such that it will be sent to the devices again. This is
typically needed when the file is modified. Normally Sync Server does not
automatically detect changes to the subscribed files. A file subscription needs
to be reset in order for the devices to receive new version of the file.
-->
<ELEMENT AlterFileSubscription (Name, FileSubscription)>
<ELEMENT FileSubscription (Name?, Description?, SubAdapter?, EncryptionLevel?,
Block?, DeviceTypes?, SourceFileName?,
TargetFileName?, Reset?)>

```

```

<!ELEMENT Reset EMPTY>

<!--'AlterGroup' tag can be used to list new settings for a synchronization
group.Using this tag, new subscription sets can be included into this group
or existing ones can be excluded from this group. Data filters can be added or
removed. The Group can be enabled or disabled. Name and description can also be
changed if needed.
-->
<!ELEMENT AlterGroup (Name, Group)>
<!ELEMENT Group (Name?, Description?, Enabled?,
                (IncludeSubscriptionSet | ExcludeSubscriptionSet
                 | AddDataFilter | RemoveDataFilter)* )>

<!ELEMENT ExcludeSubscriptionSet (Name)>

<!--To alter a data filter, remove it first and add it again.-->
<!ELEMENT RemoveDataFilter (Name) >

<!--'Alteruser' tag can be used to list new settings for a synchronization user.
Using this tag, the user can be assigned to a new group or if <NULL/> is specified
for the 'UserGroup', the user will just be removed from it's current group.

User can add or remove data filter values that it overrides. Name, password and
description can also be changed if needed.
LDAPProperties tag only applies in LDAP enabled environments.
This tag lists LDAP specific properties of a user.
-->
<!ELEMENT AlterUser (Name, User?)>
<!ELEMENT User (Name?, UserGroup?, Description?, Password?, LDAPProperties?,
               (OverrideDataFilter | RemoveDataFilter)*)>

<!ELEMENT AlterDevice (Id, Device?)>
<!ELEMENT Device (Id?, Type?, Enabled?, Version?)>

<!--'AlterSubscriptionSet' is typically used to add/remove subscriptions to a
subscription set.
-->
<!ELEMENT AlterSubscriptionSet (Name, SubscriptionSet?)>
<!ELEMENT SubscriptionSet (Name?, Description?,
                          (IncludeSubscription | ExcludeSubscription)* )>

<!ELEMENT ExcludeSubscription (Name)>

<!--'AlterJdbcSubscription' tag is typically used to add/remove tables of
a jdbc subscription. If you wish to change the mirror/master databases of
a subscription, you need to create a new susbcription instead of changing
an existing one. Note that,jdbc connection password of a database can be
changed using AlterJdbcMirror, AlterJdbcMaster tags.'AlterTable' tag can be
used to change synchrononization and replication properties of a subscribed table.
It can also be used to add/remove indexes for a table.
-->
<!ELEMENT AlterJdbcSubscription (Name, JdbcSubscription)>
  <!ATTLIST AlterJdbcSubscription Replicate (TRUE | FALSE) "TRUE">

<!ELEMENT JdbcSubscription (Name?, Description?,
                           EncryptionLevel?, Block?,
                           (AddTable | RemoveTable | AlterTable)*)>

<!ELEMENT AlterUploadSubscription (Name, UploadSubscription)>
<!ELEMENT UploadSubscription (Name?, Description?,
                              EncryptionLevel?, Block?,
                              (AddTable | RemoveTable | AlterTable)*)>

<!ELEMENT AlterDproprSubscription (Name, DproprSubscription)>
  <!ATTLIST AlterDproprSubscription Replicate (TRUE | FALSE) "TRUE">
  <!ATTLIST AlterDproprSubscription SetupDPropRSourceAndMirror

```

```

(TRUE | FALSE) "FALSE">

<!ELEMENT DpropSubscription (Name?, Description?,
                             EncryptionLevel?, Block?, CommitCount?,
                             (AddTable | RemoveTable | AlterTable)*)>

<!ELEMENT RemoveTable (TargetSchema, TargetTable)>
<!ELEMENT AlterTable (TargetSchema, TargetTable, Table)>
<!ELEMENT Table (TargetSchema?, TargetTable?, SourceSchema?, SourceTable?,
                 DeleteAuth?, InsertAuth?, UpdateAuth?, SelectAuth?,
                 WhereClauseMirrorToMobile?, WhereClauseMasterToMirror?,
                 JoinFilter?, TableSpace?, Order?, Encrypt?,
                 (AddIndex | AddCustomLogic |
                  RemoveIndex | RemoveCustomLogic)*)>

<!ELEMENT RemoveIndex (Name)>
<!ELEMENT RemoveCustomLogic (PolicyType)>

<!ELEMENT AddProperty (Type,Name,Value?,Description?)>
<!ELEMENT RemoveProperty (Type,Name)>
<!ELEMENT AlterProperty (Type,Name,Property)>
<!ELEMENT Property (Type?,Name?,Value?,Description?)>

<!--'ReplicateMirror' replicates a mirror database.
'Database' is the DB2 database name for the mirror database.
'Timeout' is the number of seconds to wait for replication to begin.
-->
<!ELEMENT ReplicateMirror (Database)>
  <!ATTLIST ReplicateMirror Timeout CDATA #IMPLIED>

<!--'PruneMirror' prunes the history tables of a mirror database.
'Database' is the DB2 database name for the mirror database. Database
is optional. User can request the pruning of all mirror databases that
are in use by specifying PruneAll="TRUE" attribute.
This attribute is false by default.
-->
<!ELEMENT PruneMirror (Database?)>
  <!ATTLIST PruneMirror PruneAll (TRUE | FALSE) "FALSE">

```

Planning and design

Topics in this section introduce the differences between the Data Propagator and JDBC replication engines. Also use these topics to learn how the two types of engines work to replicate data between enterprise servers and mobile devices.

A comparison of the Data Propagator and JDBC replication engines

Overview of DB2 Everyplace subscriptions and replication engines

A DB2 Everyplace subscription specifies how data on a source system (the enterprise server) is transferred to and from the target system (the mobile device). The DB2 Everyplace JDBC replication subscription and the Data Propagator replication subscription are powerful, flexible tools for delivering enterprise data to mobile users. A *replication engine* is a component of the DB2 Everyplace Sync Server; the replication engine is the mechanism that runs the replication subscriptions.

Replication subscriptions use two important processes to manage data: replication and device synchronization. Replication transfers data between a *source* system and a *mirror* system. Device synchronization transfers data between a mirror system and *target* system (see figure 1 below). A source system contains the data that you want to replicate. A target system is a database that resides on a mobile device and sends and receives its data from the source system. A mirror system is an internal database that stores the data that is transferred between the source system and the target system. Replication subscriptions used in conjunction with data filters and multiple mirror databases might offer better

performance than subscriptions that do not use a replication engine, such as upload subscriptions and file subscriptions. With a replication subscription, you can unite distributed relational databases to form an integrated database solution.

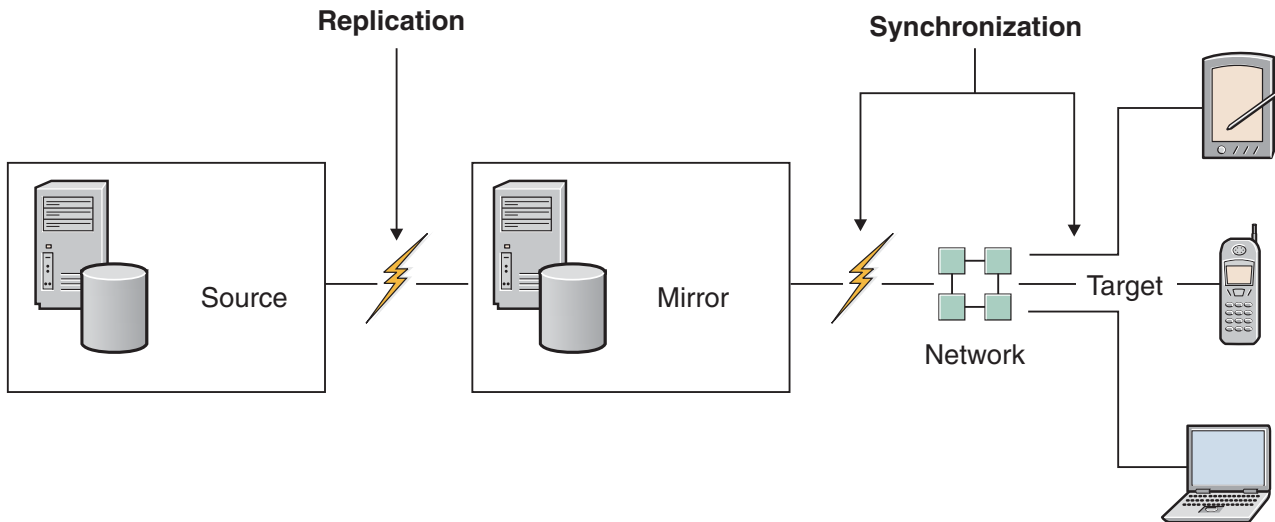


Figure 2. The replication and synchronization processes in DB2 Everyplace

DB2 Everyplace supports two replication engines:

- The DataPropagator replication engine (also known as the IBM SQL Replication)
- The JDBC replication engine

The Data Propagator replication engine uses the IBM DB2 Data Propagator Update Anywhere solution, which supports only DB2 Version 9.1 as a data source. The JDBC replication engine supports multiple data sources, including IBM DB2 Version 9.1, IBM Informix, Oracle, Microsoft, and other JDBC-compliant data sources. Each replication engine is best suited for different requirements. Normally, only one replication engine is chosen for a particular environment. You can use both types of replication engine, but only a single replication engine can be used for each mirror database.

Differences between the Data Propagator replication engine and the JDBC replication engine:

The following table provides a brief overview of differences between the Data Propagator replication engine and the JDBC replication engine.

An understanding of the replication mechanisms is useful for setting up and maintaining a DB2 Everyplace solution. Carefully test and evaluate both replication mechanisms in your environment because each environment is different and might react in unexpected ways.

Table 6. Differences between the Data Propagator replication engine and the JDBC replication engine

| Feature Overview | Data Propagator | JDBC |
|---|-----------------|------|
| Full administration support provided by the DB2 Everyplace Sync Server Administration Console GUI | No | Yes |
| Full administration support provided by XML Scripting | Yes | Yes |
| Supports any JDBC-compliant data source | No | Yes |
| Minimal performance impact to applications on the source database | Yes | No |
| Minimal performance impact on the replication process | No | Yes |

Table 6. Differences between the Data Propagator replication engine and the JDBC replication engine (continued)

| Feature Overview | Data Propagator | JDBC |
|---|-----------------|------|
| Flexibility in conflict resolution behavior | No | Yes |
| No special privileges required on source tables to create and invoke triggers | Yes | No |
| Supports referential integrity constraints and existing triggers on source tables | No | Yes |

“The Data Propagator replication engine”

The Data Propagator replication engine uses a change-capture mechanism and a log to replicate data between a source system and a target system. A *capture* process running on the source system captures changes as they occur in the source tables and stores them temporarily in the change data tables. The database administrator of the source database must ensure that the Data Propagator capture process is active on the source system. The *apply* process reads the change data tables and applies the changes to the target tables.

“The JDBC replication engine” on page 132

The JDBC replication engine uses a change-capture mechanism and triggers to replicate data between a source system and a target system. When you create a subscription, the JDBC replication engine creates triggers for each source table. The triggers capture changes as they occur in the source tables and run SQL commands to store the changes temporarily in the change data tables. The replication engine then reads the change data tables and replicates the data to the mirror tables.

Conflict resolution:

The primary key of a relational table uniquely identifies each record in a table. If changes are made by both the source and client applications to records that share the same primary key value, *conflict resolution* is invoked to determine the resultant value of the data.

The Data Propagator replication engine uses the Data Propagator standard conflict resolution. If there is a conflict between the source table and the target table, the data modified in the source table is accepted, and the data that was modified in the client table is rejected.

The JDBC replication engine uses the default conflict resolution behavior, which has the same behavior as the Data Propagator replication engine. You can choose to override this behavior by writing your own custom conflict resolution logic. For example, you can modify the conflict resolution to merge the data changes, or to select which data changes are accepted based on a criteria other than the default.

The Data Propagator replication engine

The Data Propagator replication engine uses a change-capture mechanism and a log to replicate data between a source system and a target system. A *capture* process running on the source system captures changes as they occur in the source tables and stores them temporarily in the change data tables. The database administrator of the source database must ensure that the Data Propagator capture process is active on the source system. The *apply* process reads the change data tables and applies the changes to the target tables.

You can create a Data Propagator subscription with the DB2 Everyplace XML Scripting tool. In DB2 Everyplace Version 8.2, you cannot create or configure Data Propagator subscriptions by using the DB2 Everyplace Sync Server Administration Console. You can use the DB2 Everyplace Sync Server Administration Console only to view and assign Data Propagator subscriptions to subscription sets. All the Data Propagator subscriptions use the Data Propagator replication engine.

Each Data Propagator replication environment consists of a *source system* and a *mirror system*. The source system contains the source database, the tables that you want to replicate, and the capture process that is

used to capture the data changes. The mirror system contains the mirror database and tables. DB2 Everyplace starts the apply process on the mirror system.

When capturing changes to the change data tables, the capture process that is running on both the source system and the mirror system will consume processor resources and input/output resources. As a result of this additional load on the source system, replication competes with the source applications for system resources. Additionally, with the Data Propagator engine, the number of moves that the changed data has to make between the tables in the mirror system is higher than with the JDBC engine. As a result, the mirror database requires a substantially larger logging space than the JDBC replication engine. Capacity planners should balance the needs of the replication tasks and source application to determine the size of the source system accordingly.

The XML scripting tool

What is a Data Propagator subscription?

Creating a Data Propagator subscription

This topic provides an overview of the steps that you need to perform to create DataPropagator table subscriptions.

How the Data Propagator replication engine handles data changes to the source system:

When the source application changes a table in the source system, the changes are synchronized Data Propagator replication engine first captures the changes, synchronizes them to the mirror system, and then applies them to the target system (the mobile device).

This process is illustrated in the following figure:

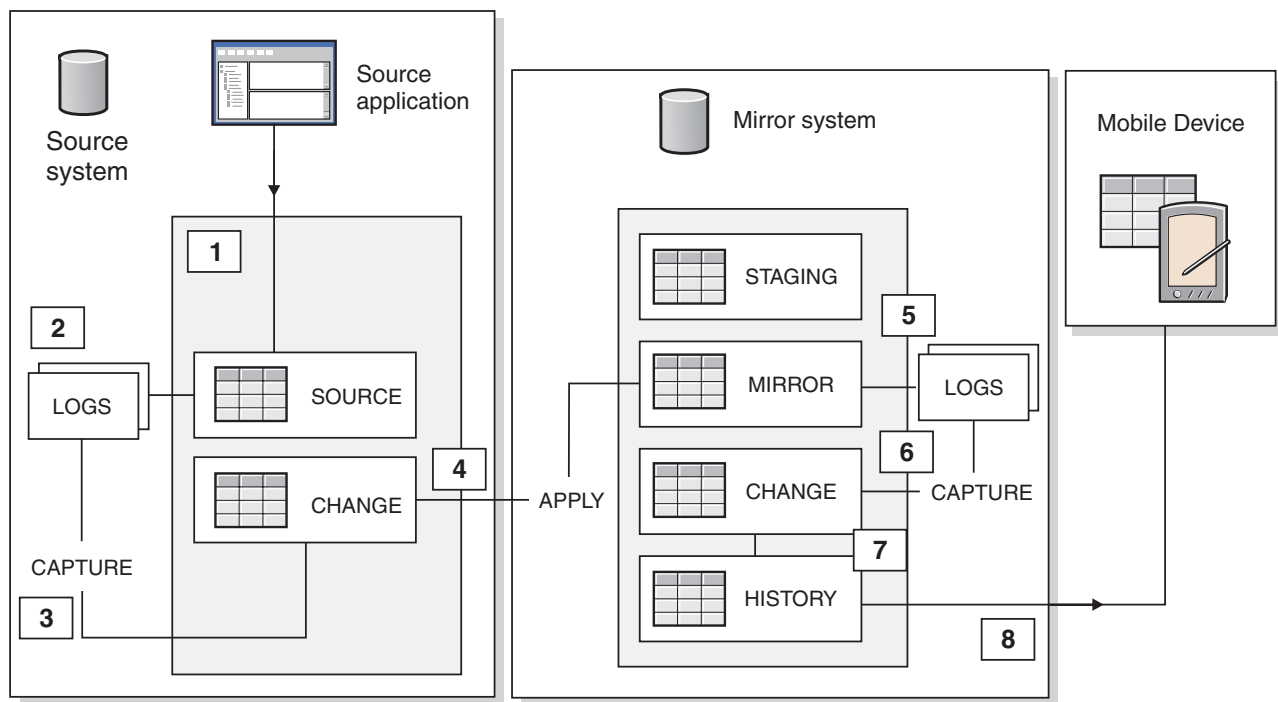


Figure 3. The process that the Data Propagator replication engine uses to handle changes to the source system

1. The source application makes changes to the source tables.
2. The DB2 transaction log saves the changes.
3. The capture process on the source system asynchronously reads the changes from the logs and captures them to the change data tables.

4. The Data Propagator apply process reads the changes from the change data tables and applies them to the mirror tables on the mirror system. There is no lock activity on the source tables during this process.
5. The DB2 transaction logs saves these changes.
6. The capture process on the mirror system reads the changes from the logs and captures the changes to the change data tables. DB2 Everyplace ensures that the capture process on the mirror system is run asynchronously at the start of replication.
7. The replication engine reads from the change data tables and updates the changes to the history tables.
8. The next time a client synchronizes, the changes are sent from the history tables to the client tables.

How the Data Propagator replication engine handles data changes to the client system:

When the client application on the mobile device changes a table in the client system, the Data Propagator replication engine first synchronizes the changes to the mirror system, captures them, and then applies them to the source system.

This process is illustrated in the following figure:

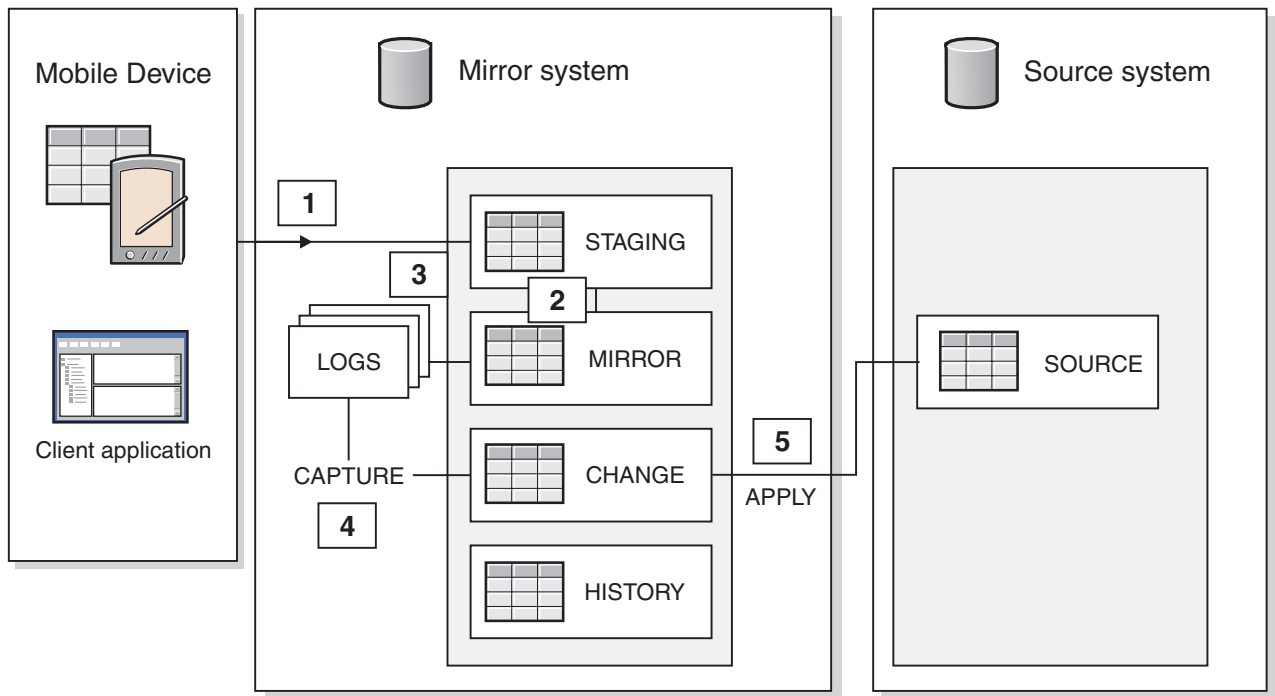


Figure 4. The process that the Data Propagator replication engine uses to handle changes to the client system

1. The client application makes changes to the client tables. The changes that are synchronized in request messages are processed to the staging tables in the mirror system.
2. From the staging tables, the changes are prepared and sent to the mirror tables.
3. The DB2 transaction logs save the changes.
4. The capture process on the mirror system reads the changes from the logs and captures them to the change data tables.
5. The apply process reads the changes from the change data tables and applies them to the source tables on the source system.

The JDBC replication engine

The JDBC replication engine uses a change-capture mechanism and triggers to replicate data between a source system and a target system. When you create a subscription, the JDBC replication engine creates triggers for each source table. The triggers capture changes as they occur in the source tables and run SQL commands to store the changes temporarily in the change data tables. The replication engine then reads the change data tables and replicates the data to the mirror tables.

You can create a JDBC subscription with the DB2 Everyplace XML Scripting tool or with the DB2 Everyplace Sync Server Administration Console. All the JDBC subscriptions use the JDBC replication engine.

Each JDBC replication environment consists of a *source system* and a *mirror system*. The source system contains the source database, the tables that you want to replicate, and the triggers that are used to capture the data changes. The mirror system contains the mirror database and the mirror tables.

When the source application makes insert changes, update changes, or delete changes to the source tables, the triggers are automatically activated to capture the changes. The trigger actions will increase the time that it takes for an application to make changes to the source tables that contain the triggers. This immediate performance impact to the source applications should be carefully considered prior to deployment on a production system.

“Mobile Devices Administration Center” on page 9

This topic describes the Mobile Devices Administration Center, a tool you can use to manage access to applications and data.

“What is a JDBC subscription?” on page 27

This topic introduces the concept of *JDBC subscriptions*.

“Creating a JDBC subscription” on page 33

Use the **Create JDBC Subscription** window to create a JDBC table subscription. Each table subscription that you create can reference one or more tables.

How the JDBC replication engine handles data changes to the source system:

When a source application makes changes to a table in the source system, the JDBC engine first captures the changes to the source system, synchronizes them to the mirror system, and then applies them to the target system (the mobile device).

This process is illustrated in the following figure:

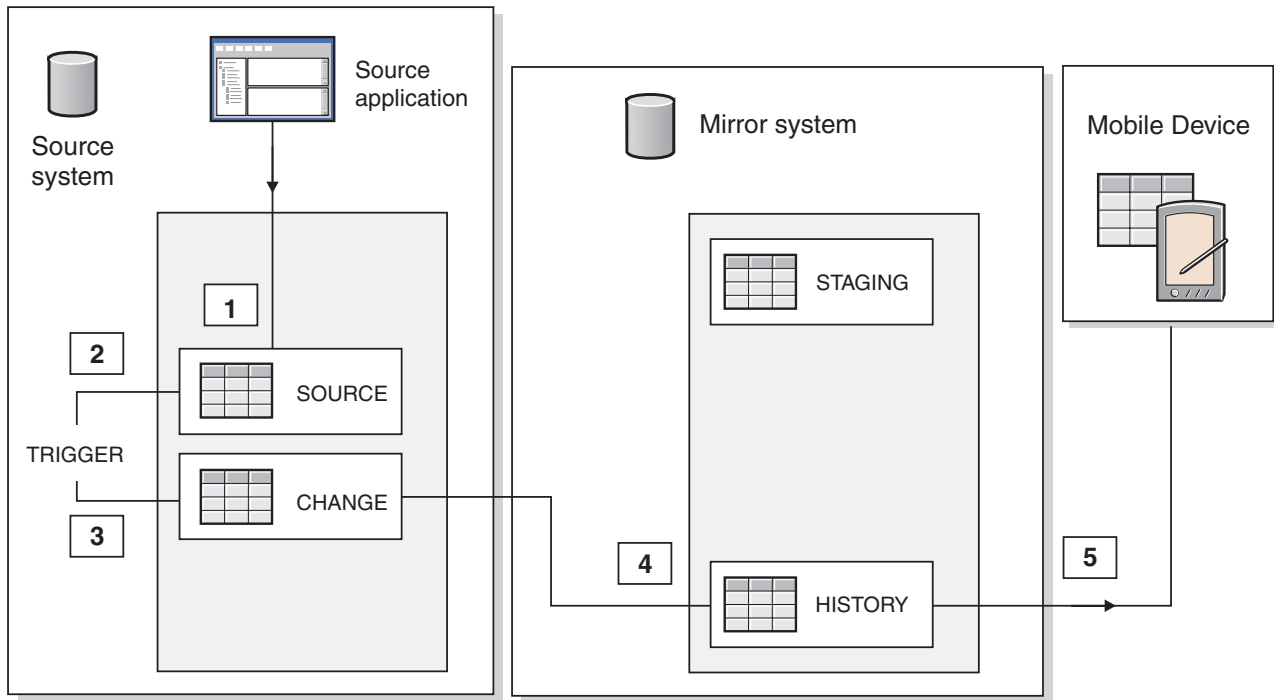


Figure 5. The process that the JDBC replication engine uses to handle changes to the source system

1. The source application makes changes to the source tables.
2. The source system activates the triggers. The triggers capture the changes to the source data tables.
3. The JDBC replication engine reads the changes from the change data tables and applies them directly to the history tables on the mirror system.
4. The next time that a client synchronizes, the changes are sent from the history tables on the mirror system to the client tables.

When the JDBC replication engine reads from the change data table, it places a short-lived, shared lock on the source table to obtain a snapshot of the data as it appears at that specific point of time. This table lock is maintained while the engine creates the snapshot, which typically takes less than one second. If the replication engine is blocked by another application that is updating the source table, the lock might appear longer because the replication engine is waiting for the lock. The replication engine will wait until the wait time that has been configured for the table has passed. While the replication engine is holding or waiting for the lock, other source applications will not be able to update the source table until the lock is released. If the replication engine does not obtain the lock after the specified wait time has passed, the engine will receive a timeout error from the data source to terminate the replication process, and recovery will be performed at the next replication cycle.

How the JDBC replication engine handles data changes to the client system:

When the client application (the mobile device) changes a table in the client system, the changes are synchronized from the client system to the mirror system. After the changes have been synchronized, the JDBC replication captures the changes on the mirror system.

This process is illustrated in the following figure:

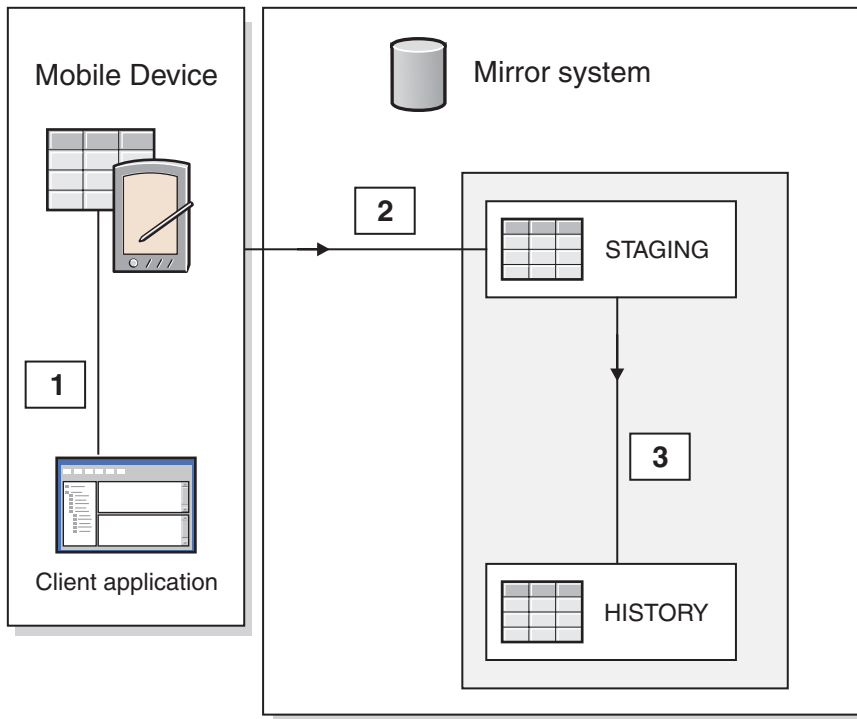


Figure 6. The process that the JDBC replication engine uses to handle changes to the client system

1. The client application changes a client table.
2. The changes are synchronized in request messages and sent to the staging tables in the mirror system.
3. The history tables on the mirror system are updated.

The changes are applied in batches, which can cause the replication engine to deadlock with the source applications. When a database discovers a deadlock, it picks a *victim*, and then rolls back the changes made by the victim. If the replication engine is involved in a deadlock situation with a source application and it is chosen as the victim, the engine will receive a deadlock error from the data source, terminate processing, and recover at the next replication cycle. You should design and write source applications to avoid deadlock situations and to effectively handle deadlock situations should they occur. To minimize the potential for deadlock situations, you can configure replication to apply changes in smaller batches, or schedule replication to run when source applications are not modifying source tables.

Changing the batch size for the JDBC replication engine:

To minimize the possibility of deadlock, you can schedule the replication process to apply changes to the source table in smaller batches.

To change the batch size, open a command window and switch to the Server/bin or Server\bin directory within the directory where DB2 Everyplace is installed. Run the following command:

```
dsysetproperty DSYGdf1t Jdbc.Replication.OperationsPerTransaction = x
```

where *x* is the number of changes that the replication engine should apply to a source table in one transaction.

Synchronizing databases in DB2 Everyplace

Synchronizing is the process for users to access their organization's data locally on a mobile device, modify this data, and reconcile the changes with a database on a remote server in a timely fashion.

The synchronization process

Mobile workers require a reliable software solution that allows them to access their organization's data locally on a mobile device, modify this data, and synchronize the changes with a database on a remote server in a timely fashion. IBM's DB2 Everyplace solution provides this capability, allowing two-way synchronization of data between the enterprise data source and mobile and embedded devices. DB2 Everyplace is also capable of one-way synchronization of files. DB2 Everyplace Sync Server can also manage one-way subscriptions where DB2 Everyplace only inserts data into the data source.

The synchronization process has two major steps:

- Mobile users submit changes that they made to local copies of source data.
- Users receive any changes that were made to source data residing on the enterprise server since the last time they synchronized.

This two-step process is known as a *synchronization session*.

Related concepts

"Synchronization from the mobile device to the source database" on page 136

This topic shows how changes that a user submits for synchronization are applied to the source database during synchronization.

"Synchronization from the source database to the mobile device" on page 142

This topic shows how changes from a source table are applied to a DB2 Everyplace table on the user's mobile device during synchronization.

"Synchronization conflict resolution" on page 157

"Components of the DB2 Everyplace solution" on page 4

DB2 Everyplace Enterprise Edition is a robust solution for synchronizing enterprise data. You can configure the DB2 Everyplace environment in multiple ways depending on the needs of your network and your users.

"DB2 Everyplace environments" on page 5

DB2 Everyplace Enterprise Edition is a robust solution for synchronizing enterprise data. You can configure the DB2 Everyplace environment in multiple ways depending on the needs of your network and your users.

Enabling and disabling synchronization

Before synchronization can happen, both the user and group must be enabled in the Mobile Devices Administration Center. If you delete required information from an object at a later time, the object is automatically disabled.

Prerequisite:

Create the user in the Mobile Devices Administration Center before his or her device is registered during the first synchronization. See "Allowing multiple devices per user" on page 21 for information about letting one user synchronize multiple mobile devices.

- You can enable a group for synchronization after you assign to the group:
 - A unique name

- At least one user
- At least one subscription set
- A user is automatically enabled for synchronization after you:
 1. you provide a unique name for the user.
 2. you assign the user to an enabled group.
 3. the user's device is registered.

Related tasks

“Enabling synchronization” on page 26

This topic introduces two approaches you can use to enable synchronization.

“Disabling synchronization” on page 19

You might choose to disable an object temporarily if you are making administration changes to either the group or the user. This topic introduces how to disable

Synchronizing mobile devices

This topic explains how to synchronize data from a mobile device to a database on a server.

Synchronization from the mobile device to the source database

This topic shows how changes that a user submits for synchronization are applied to the source database during synchronization.

The numbers in the figure correspond to the explanation following it:

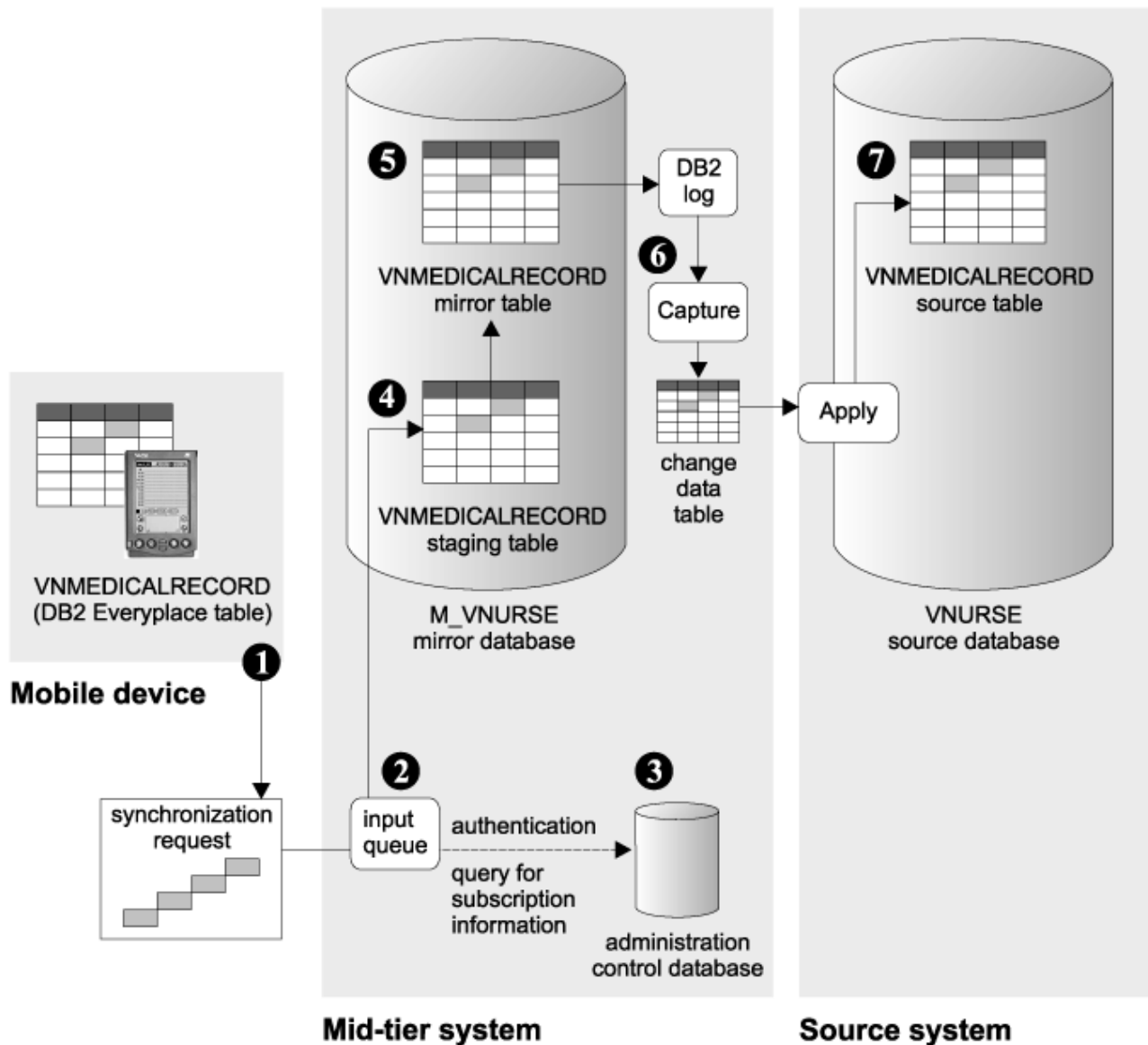


Figure 7. How changes that a user submits for synchronization are applied to the source database

1. A home health care specialist or visiting nurse updates the blood pressure of a patient in a local copy of the VNMEDICALRECORD table on a Palm OS device. The nurse exits the application used to edit the table, then taps the **DB2 Sync** icon to start the synchronization client software on the device. A mobile application can also be written to include the ability to start the synchronization, utilizing the DB2 Everyplace Sync Client API. When the synchronization client application starts, the nurse chooses the name of the application to synchronize, then taps **Synchronize** to request synchronization.
2. The request is authenticated and then placed on an input queue on the mid-tier system. The synchronization client software on the device waits for a synchronization reply from the source server (see “Synchronization from the source database to the mobile device” on page 142).
3. Users can synchronize only the subset of data and files to which they have been subscribed.
4. The data is placed into a staging table. Staging tables help improve throughput capacity of synchronization requests because changes can be staged while other updates are taking place.
5. The data is copied from the staging table to the mirror table (VNMEDICALRECORD in this example) and potential update conflicts are resolved. Changes to the mirror table are recorded in the DB2 log.

6. The DB2 DataPropagator Capture program starts. This program captures changes to the mirror table from the DB2 log and writes them to a change data (CD) table.
7. The DB2 DataPropagator Apply program starts and applies changes from the CD table to the source table, VNMEDICALRECORD, in the VNURSE database on the source system.

Related concepts

“Synchronization from the mobile device to the source database” on page 136

This topic shows how changes that a user submits for synchronization are applied to the source database during synchronization.

“Synchronization from the source database to the mobile device” on page 142

This topic shows how changes from a source table are applied to a DB2 Everyplace table on the user’s mobile device during synchronization.

“Synchronization conflict resolution” on page 157

“Components of the DB2 Everyplace solution” on page 4

DB2 Everyplace Enterprise Edition is a robust solution for synchronizing enterprise data. You can configure the DB2 Everyplace environment in multiple ways depending on the needs of your network and your users.

“DB2 Everyplace environments” on page 5

DB2 Everyplace Enterprise Edition is a robust solution for synchronizing enterprise data. You can configure the DB2 Everyplace environment in multiple ways depending on the needs of your network and your users.

Synchronizing using DB2 Sync

This topic explains how to set up the necessary software and mobile devices in order to test synchronization using DB2 Sync.

The DB2 Sync sample application

The DB2 Sync sample application works with the DB2 Everyplace Sync Server to synchronize data and applications between mobile devices and enterprise data sources. Before you configure DB2 Sync, you need to define a user, group, subscription, and subscription set in the Mobile Devices Administration Center.

DB2 Sync is a synchronization application with a graphical interface. It is available for Palm OS, Windows CE, and the Symbian OS mobile devices. DB2 Sync Console, a command-line program, is also provided to test synchronization on Windows, Neutrino, and Linux mobile devices. Both the applications are open source and demonstrate how to use the DB2 Everyplace Sync Client C-API.

Important: Read the license agreement before you use the source code.

During every synchronization, the client software checks whether the client’s configuration (the subscription sets and subscriptions that are assigned to the client’s group) must be updated. For example, when you refresh a file that is referenced in a subscription, DB2 Everyplace sets a flag that indicates that the subscription has changed. The updated subscription is downloaded to all subscribed users on the next synchronization.

Configuring DB2 Sync for a Palm OS device

After you set up DB2 Sync, you must configure the DB2 Sync application in order to perform synchronization. This topic describes the configuration for the DB2 Sync application.

Prerequisite: Be sure that you have installed DB2Everyplace on the mobile device and that the mobile device or emulator is set up for network connections. For more information, see *Installation and User’s Guide*.

After you set up DB2 Sync, you can configure DB2 Sync menu options. In the upper left-hand corner of the DB2 Sync dialog, click the **DB2 Sync** menu. A drop-down menu appears with the following options:

- Subscription sets
- Server settings
- Client settings
- Network settings
- About DB2 sync

Among the above configuration options, the Server settings field is necessary to successful synchronization, while the others are optional. Thus, you must configure the server settings.

Configuring Server settings on DB2 Sync for a Palm OS device:

This topic describes how to configure the Server settings field that is necessary for successful synchronization.

Be sure that you have installed DB2 Everyplace on the mobile device and that the mobile device or emulator is set up for network connections. For more information, see *Installation and User's Guide* and the DB2 Sync menu options topic.

1. Locate the DB2 Sync application on the mobile device or emulator.
2. Start **DB2 Sync**.
3. Select **Server Settings** from the menu.
4. Tap the drop-down list next to URL.
5. Select **-enter-** and enter the DB2 Everyplace Sync Server URL. The default port used by the DB2 Everyplace Sync Server basic application server is 8080. If the DB2 Everyplace Sync Server is configured on a port other than 8080, you must specify the port number in the URL. Each new server URL that you enter is stored in the drop-down list.
6. Type a user ID in the **User** field. The user name that you enter must be defined in the DB2 Everyplace Mobile Devices Administration Center. For information about creating a user in the Mobile Devices Administration Center, see *Sync Server Administration Guide*.
7. Type a password in the **Password** field. The password that you enter must be defined in the DB2 Everyplace Mobile Devices Administration Center. For information about creating a user in the Mobile Devices Administration Center, see *Sync Server Administration Guide*.
8. Select the **Save Password** check box if you want to save the password for subsequent uses.

DB2 Sync menu options:

This topic describes the menu options for configuring DB2 Sync.

In the upper left-hand corner of the DB2 Sync dialog, click the **DB2 Sync** menu. A drop-down menu appears with the following options:

- Subs sets
- Server settings
- Client settings
- Network settings
- About DB2 sync

Subscription sets

Note: Configuring subscription sets is optional for successful synchronization.

You can view the subscription sets to which the DB2 Everyplace Sync Client subscribes by selecting the **Subscription Sets** option from the menu. In the subscription set panel, the check box next to a

subscription set indicates whether the subscription set is enabled for synchronization. Thus, you can disable synchronization on subscription sets by clearing a check box. The command buttons and their actions in the panel are explained as follows:

OK After changing the synchronization options of the subscriptions, you can save the changes by clicking **OK**.

Cancel

If you want to discard the changes, click **Cancel**.

Details

When you click the **Details** button, you can see the detailed information about the selected subscription set. In addition, if you want to discard the mobile device data and perform a refresh of a subscription set, you can do this by selecting the **Reset** check box. The next time that you synchronize, the DB2 Everyplace Sync Client refreshes the data on that subscription set by dropping the data and re-fetching the source data from the server.

Purge This button drops the current subscription set information. The next time that you synchronize, the DB2 Everyplace Sync Client refreshes the subscription set information, and then refreshes each subscription set.

Important: If you made changes to the local tables but have not synchronized these changes with the source database, you will lose those changes when you perform a purge operation.

Server settings

Note: Configuring server settings is necessary for successful synchronization.

When you click the **Server Settings** menu option, the Setting dialog opens. In this dialog, you can configure the URL, user ID and password of the DB2 Sync settings.

Client settings

Note: Configuring client settings is optional for successful synchronization.

You can configure the settings that are specific to the client by selecting the **Client Settings** option from the menu. The client settings that can be configured are as follows:

Trace Tap the drop-down list next to **Trace**, and select **Detailed** to generate detailed tracing information when you encounter problems during synchronization.

Memory slot

Tap the drop-down list next to **Memory slot**, and select the target memory expansion card where you want to save the client information and synchronized data. The default **Device** setting is the main memory on the mobile device. When a memory slot other than **Device** is selected, you can also specify the exact target path in the **Target Path** field, for which the default is the root.

Table 7 lists the default paths where DB2 Sync stores client information and synchronized data for each platform.

Table 7. Default paths for DB2 Everyplace Sync Client information and synchronized data

| Platform | Default installation path |
|--------------|---------------------------|
| Palm | Main memory |
| Windows | Current directory |
| Linux | Current directory |
| QNX Neutrino | Current directory |

Table 7. Default paths for DB2 Everyplace Sync Client information and synchronized data (continued)

| Platform | Default installation path |
|------------|---------------------------|
| Windows CE | \ |
| Symbian | C:\System\Data\Sync\ |

Network settings

Note: Configuring network settings is optional for successful synchronization.

You can configure the network settings by selecting the **Network Settings** option from the menu. The network settings that can be configured are as follows:

Timeout

In this drop-down list, you can specify a timeout duration for synchronization. Select a longer duration if the server is busy or if there is a large amount of data to be synchronized. If you frequently encounter timeout error messages during synchronization, contact the system administrator for the proper setting of this value. The default value is 1 minute.

Network Speed

Tap the drop-down list next to **Network Speed**, and select the proper network speed. This setting allows the DB2 Everyplace Sync Client to adjust the actual message size when communicating with the DB2 Everyplace Sync Server to achieve the best network performance.

Use Proxy

If the system uses a proxy server, select the **Use Proxy** check box and enter the IP address and the port number of the proxy server.

About DB2 Sync

When you select **About DB2 Sync**, DB2 Sync displays the mobile device information, including the version number and the build date for the DB2 Everyplace client and device ID.

Synchronizing data using DB2 Sync

Prerequisites:

Before you use the DB2 Sync application, you need to configure the DB2 Sync application as described in “Configuring Server settings on DB2 Sync for a Palm OS device” on page 139.

1. To synchronize data using DB2 Sync:

- a. Start the mobile device.
- b. Start the DB2 Sync application. The system displays the main DB2 Sync window.
- c. Tap **Synchronize**. The synchronization begins. The panel will display the status and the progress of the synchronization. You can cancel a synchronization at any time by clicking the **Cancel** button. When the synchronization ends, the following messages will appear to indicate the synchronization succeeds, fails, or get cancelled respectively.
 - Synchronization succeeded
 - Synchronization failed
 - Synchronization cancelled

If the synchronization fails, you can click the **Log** button to see the cause (error) of the failure. If the synchronization is successful, then you can go on to verify that the synchronized data is correct.

2. Verify the data on the client.

- a. Start DB2eCLP on the mobile device.

- b. Enter an SQL statement that selects all the records from the table that you recently synchronized.
3. Verify the data on the data source using the DB2 Command Line Processor:
 - a. Open the DB2 CLP on the source database.
 - b. Enter an SQL statement to select all the records of the subscribed table.

Note: If you are using the standalone version of DB2 Version 9.1, you can also verify the data on the source database using the DB2 Version 9.1 Control Center. Right-click the subscribed table and select **Sample Contents** to browse the contents of the table.

4. Compare the contents of the table on the mobile device to that of the data source.

Related concepts

“The DB2 Sync sample application” on page 138

“DB2 Sync menu options” on page 139

This topic describes the menu options for configuring DB2 Sync.

Related tasks

“Configuring Server settings on DB2 Sync for a Palm OS device” on page 139

This topic describes how to configure the Server settings field that is necessary for successful synchronization.

Synchronization on the server

This topic presents information about server-side tasks and issues related to synchronizing data.

Synchronization from the source database to the mobile device

This topic shows how changes from a source table are applied to a DB2 Everyplace table on the user’s mobile device during synchronization.

The DB2 Everyplace Sync Server sends to the user all relevant source data changes that have been made since the user’s last synchronization. The DB2 Everyplace Sync Server sends only changed data that the user is subscribed to.

The numbers in the figure correspond to the explanations following it.

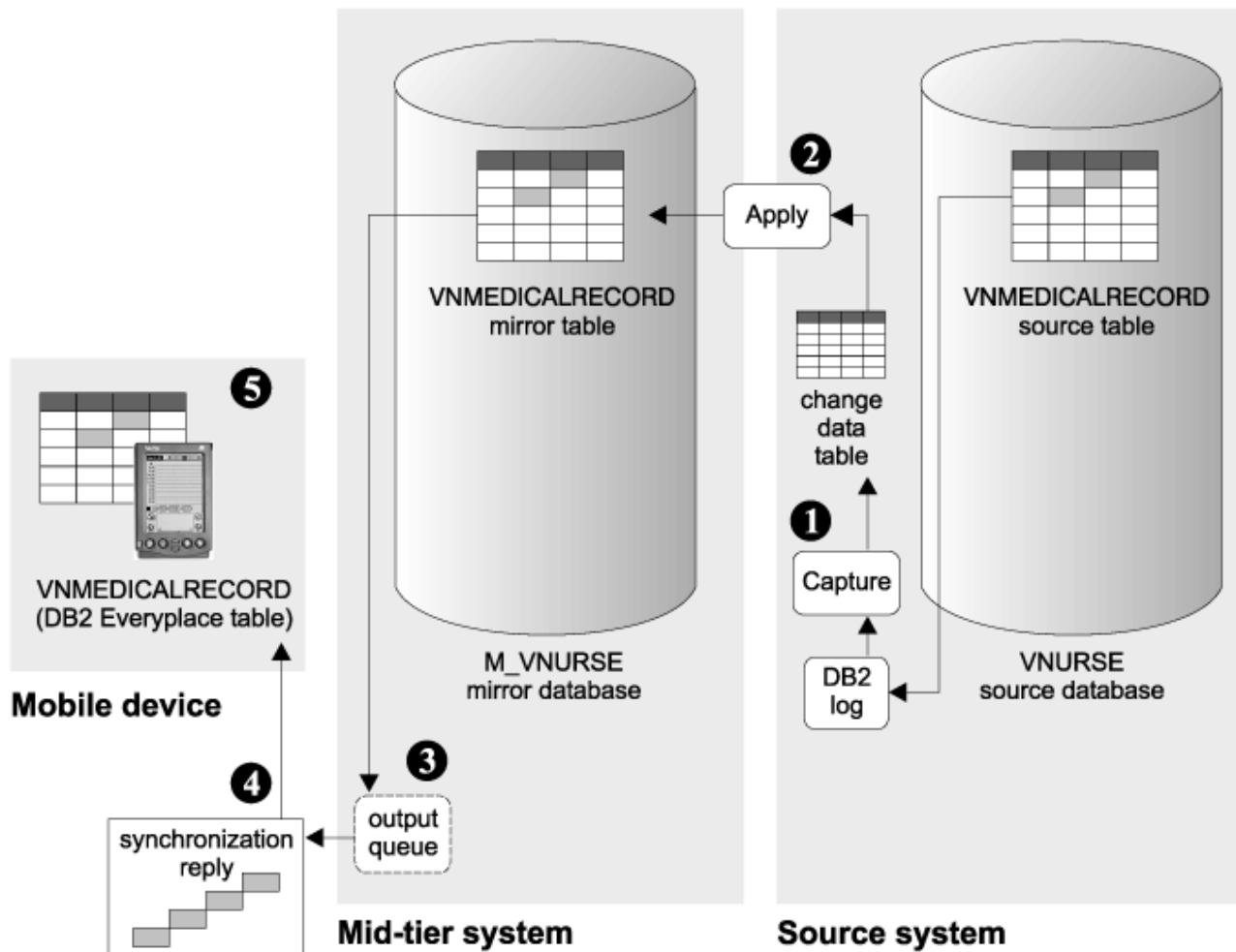


Figure 8. How changes from the source database are applied to the mobile database

1. The DB2 DataPropagator Capture program, running continuously on the source system, captures changes to the VNMEDICALRECORD source table from the DB2 log and writes them to the CD table.
2. The DB2 DataPropagator Apply program applies changes from the CD table to the mirror table, VNMEDICALRECORD. If update conflicts are identified they are resolved during this step of the process.
3. The changes to data are sent to an output queue on the mid-tier system as a synchronization reply message.
4. The synchronization client software retrieves the synchronization reply message from the output queue. The client retrieval request is authenticated.
5. The changes to data are applied to the client's local copy of the table. The synchronization session ends.

Related concepts

"Synchronization from the mobile device to the source database" on page 136

This topic shows how changes that a user submits for synchronization are applied to the source database during synchronization.

"Synchronization from the source database to the mobile device" on page 142

This topic shows how changes from a source table are applied to a DB2 Everyplace table on the user's mobile device during synchronization.

"Synchronization conflict resolution" on page 157

“Components of the DB2 Everyplace solution” on page 4

DB2 Everyplace Enterprise Edition is a robust solution for synchronizing enterprise data. You can configure the DB2 Everyplace environment in multiple ways depending on the needs of your network and your users.

“DB2 Everyplace environments” on page 5

DB2 Everyplace Enterprise Edition is a robust solution for synchronizing enterprise data. You can configure the DB2 Everyplace environment in multiple ways depending on the needs of your network and your users.

Synchronization tutorial

After installing the DB2 Everyplace database and the DB2 Everyplace Sync Server and setting up your device, you are ready to synchronize. The following topics test synchronization using sample data and sample DataPropagator and JDBC subscriptions, but does not use sample upload subscription types.

This tutorial assumes understanding of database administration for DB2 Version 9.1 and JDBC-compliant databases.

Notes:

- The source tables specified in a DataPropagator subscription must be on a DB2 Version 9.1 database.
- A DataPropagator subscription cannot use the same mirror database as one used by a JDBC subscription.
- For a JDBC subscription, the source tables can be on any JDBC-compliant database, including DB2 Version 9.1.

For more information about working with different data sources, see “Configuring the DB2 Everyplace Sync Server to access data sources using JDBC” on page 30.

Creating a JDBC subscription to access source data

This task is part of the greater task of setting up synchronization. When you have completed the steps below, return to “Synchronization tutorial.”


To create a JDBC subscription:

Perform the following steps in the Mobile Devices Administration Center.

1. Open the Mobile Devices Administration Center and DB2 Control Center:
 - For Windows, select **Start** → **Programs** → **IBM DB2 Everyplace** → **Start Mobile Devices Administration Center**.
 - For UNIX, change the directory to \$DSYINSTDIR/Server/bin, then run dsyadmin.sh.
2. Right-click on the Subscriptions folder in the Mobile Devices Administration Center. Select **Create** → **Table subscription** → **JDBC Subscription**.
3. Type a name for the JDBC subscription in the **Name** field. This example uses SCHEDULE.
4. Optional: Type a description for the JDBC subscription in the **Description** field. This example uses VNSchedule test subscription
5. Select the encryption level for the JDBC subscription in the **Encryption level** field. This example uses **None**
6. On the Source tab, identify the name of the source database from which data is to be synchronized.
7. Type the source database URL in the **Database URL** field. This example uses jdbc:db2:vnurse

8. Type a valid source database user ID and password in the **User ID**, **Password**, and **Verify password** fields. It is required that the same user ID and password is used for each DB2 Everyplace Sync Server subscription that connects to a single source database if multiple subscriptions are connecting to the same database.
9. Click the **Test Connection** push button to test the connection to the JDBC source database.
10. Click **Close**.
11. On the Mirror tab, identify the mirror database that the DB2 Everyplace Sync Server will use to cache data.

Note: A JDBC subscription cannot use the same mirror database as one used by a DataPropagator subscription.

12. Click the  push button next to the **Database URL** field and select **M_VN2** in the Select Mirror Database window. Click **OK**.
13. Type a valid mirror database user ID and password in the **User ID**, **Password**, and **Verify password** fields. This example uses user ID db2admin.
14. Click the **Test Connection** push button to test the connection to the JDBC mirror database. Click **Close** on the DB2 Message.
15. Click the **Identification** tab.
16. Select the **Define Subscription** push button. The Define Replication Subscription window opens.
17. In the Define Replication Subscription window, click **Add**. The Add Table window opens.
18. Identify the tables in the source database whose content will be synchronized. From the Table list of the Add Table window, select the replication source to add to this subscription. To change the source filter used to limit the available tables in the table list, click **Filter**.

Note: If a source table of a table subscription is to be dropped and re-created, the administrator must first delete that table from the table subscription, and then add that table back to the table subscription after the source table is re-created.

19. Select DSYSAMPLE.VNSCHEDULE from the Table list. The **Target schema** and **Target table** fields are automatically completed.
20. Optional: If you want to include the constraints and indexes from the source tables, select **Use Constraints defined in DB2 source** and **Use indexes defined in DB2 source**.
21. Click **Add**. Click **Close**. Return to the Define Replication Subscription window.
22. Optional: To add a primary key:
 - a. Highlight one or more tables and click **Advanced ...**. The Advanced Replication Subscription window opens, displaying the highlighted tables.
 - b. On the Target Columns tab, select the checkbox in the **Primary key** column next to the table that you want to have a primary key.
 - c. Click **OK**. Return to the Define Replication Subscription window.
23. Click the **Timing** push button in the Define Replication Subscription window.
24. In the **Batch Window** field, specify the amount of time, in seconds, to wait before starting replication. For example, if you type 60, changes will be replicated to the source database after with intervening intervals of 60 seconds. Click **OK**.
25. Click **OK** to close the Define Replication Subscription window.
26. Click **OK** to close the Create JDBC Subscription window. If the **OK** button is grayed out, all of the fields have not been filled in correctly.

DB2 Everyplace Sync Server is now configured to synchronize the data in this table using JDBC. Next you will create a DataPropagator subscription.

Related tasks

“Synchronization tutorial” on page 144

After installing the DB2 Everyplace database and the DB2 Everyplace Sync Server and setting up your device, you are ready to synchronize. The following topics test synchronization using sample data and sample DataPropagator and JDBC subscriptions, but does not use sample upload subscription types.

“Creating a JDBC subscription to access source data” on page 144

“Grouping subscriptions into mobile subscription sets” on page 56

“Creating mobile users” on page 22

You need to create a user ID and password for mobile users in order to connect the mobile device to the DB2 Everyplace Sync Server. This topic introduces how you create mobile users in the Mobile Devices Administration Center.

“Creating a data synchronization group”

“Starting the application server” on page 147

Creating a data synchronization group

With DB2 Everyplace Sync Server, you can use groups to manage users or to combine users and groups with subscription sets.

This task is part of the greater task of Setting up Synchronization. When you have completed the steps below, return to “Synchronization tutorial” on page 144.

Perform the following steps in the Mobile Devices Administration Center.

To create a data synchronization group:

1. Right-click on the **Groups** folder. Select **Create**.
2. In the Create Group window, type a name for the group in the **Name** field. This example uses Nurse.
3. Optional: Type a description for the group in the Description field. This example uses Nurses Test Group.
4. Click the **Users** tab.
5. Select the user you created from the **Available users** list. This example uses testuser.
6. Click the > push button to move the user to the **Selected users** list.
7. Click the **Subscription Sets** tab.
8. From the **Available subscription sets** list, select a subscription set. The subscription set must be created before you can assign groups to it. This example uses VNurse.
9. Click the > push button to move the subscription set to the **Selected subscription sets** list.
10. Complete the Identification, Clients, and Subscription sets tabs of the Create Group window.
11. Select **Enable Synchronization**.
12. Click **OK** to close the Create Group window.
13. Now you are ready to begin synchronizing data.

Related tasks

“Synchronization tutorial” on page 144

After installing the DB2 Everyplace database and the DB2 Everyplace Sync Server and setting up your device, you are ready to synchronize. The following topics test synchronization using sample data and sample DataPropagator and JDBC subscriptions, but does not use sample upload subscription types.

“Creating a JDBC subscription to access source data” on page 144

“Grouping subscriptions into mobile subscription sets” on page 56

“Creating mobile users” on page 22

You need to create a user ID and password for mobile users in order to connect the mobile device to the DB2 Everyplace Sync Server. This topic introduces how you create mobile users in the Mobile Devices Administration Center.

“Starting the application server”

Starting the application server

This task is part of the greater task of setting up synchronization. When you have completed the steps below, return to “Synchronization tutorial” on page 144.

- If you are using the default application server included with DB2 Everyplace Sync Server:
 1. Start the DB2 Everyplace Sync Server servlet:
 - Windows systems:**

Select **Start** → **Programs** → **IBM DB2 Everyplace** → **Start Servlet for Sync Server**. A command prompt window opens that provides status messages from the default application server.
 - Linux and UNIX systems:**
 - a. Log in as the DB2 Everyplace Sync Server instance.
 - b. Change to \$DSYINSTDIR/Server/bin directory.
 - c. Run the dsysync.sh script.
 2. Verify that the DB2 Everyplace Sync Server servlet is running correctly by going to <http://localhost:8080/db2e/db2erdb>. Check for the following message: DB2e SyncServer (current date and time)
- If you are using the embedded Websphere Application Server and want to manage the server as a system service:
 1. Login as the root user.
 2. Start the network and databases and make them accessible before starting the server. Please consult your system and database administrator to ensure the databases are available for use before starting the server.
 3. Start the embedded WebSphere Application Server:
 - On Windows, go to **Control Panel** → **Administrative Tools** → **Services**, right-click IBM WebSphere Application Server V6 - IBMDB2eServer and select **Start**.
 - On AIX®, enter the following command:

```
dsysync_<dsy_instance_name> start
```

where *<dsy_instance_name>* is the DB2 Everyplace instance name. The script is located in inittab.
 - On Linux, to enable and start the server, enter the following commands while in the /etc/init.d/ directory:

```
chkconfig dsysyncd_<dsy_instance_name> on  
dsysync_<dsy_instance_name> start
```

where *<dsy_instance_name>* is the DB2 Everyplace instance name.

Related tasks

“Synchronization tutorial” on page 144

After installing the DB2 Everyplace database and the DB2 Everyplace Sync Server and setting up your device, you are ready to synchronize. The following topics test synchronization using sample data and sample DataPropagator and JDBC subscriptions, but does not use sample upload subscription types.

“Creating a JDBC subscription to access source data” on page 144

“Grouping subscriptions into mobile subscription sets” on page 56

“Creating mobile users” on page 22

You need to create a user ID and password for mobile users in order to connect the mobile device to the DB2 Everyplace Sync Server. This topic introduces how you create mobile users in the Mobile Devices Administration Center.

“Creating a data synchronization group” on page 146

Configuring DB2 Sync for a Palm OS device

After you set up DB2 Sync, you must configure the DB2 Sync application in order to perform synchronization. This topic describes the configuration for the DB2 Sync application.

Prerequisite: Be sure that you have installed DB2Everyplace on the mobile device and that the mobile device or emulator is set up for network connections. For more information, see *Installation and User's Guide*.

After you set up DB2 Sync, you can configure DB2 Sync menu options. In the upper left-hand corner of the DB2 Sync dialog, click the **DB2 Sync** menu. A drop-down menu appears with the following options:

- Subscription sets
- Server settings
- Client settings
- Network settings
- About DB2 sync

Among the above configuration options, the Server settings field is necessary to successful synchronization, while the others are optional. Thus, you must configure the server settings.

Synchronizing data using DB2 Sync

Prerequisites:

Before you use the DB2 Sync application, you need to configure the DB2 Sync application as described in “Configuring Server settings on DB2 Sync for a Palm OS device” on page 139.

1. To synchronize data using DB2 Sync:

- a. Start the mobile device.
- b. Start the DB2 Sync application. The system displays the main DB2 Sync window.
- c. Tap **Synchronize**. The synchronization begins. The panel will display the status and the progress of the synchronization. You can cancel a synchronization at any time by clicking the **Cancel** button. When the synchronization ends, the following messages will appear to indicate the synchronization succeeds, fails, or get cancelled respectively.
 - Synchronization succeeded
 - Synchronization failed
 - Synchronization cancelled

If the synchronization fails, you can click the **Log** button to see the cause (error) of the failure. If the synchronization is successful, then you can go on to verify that the synchronized data is correct.

2. Verify the data on the client.

- a. Start DB2eCLP on the mobile device.
- b. Enter an SQL statement that selects all the records from the table that you recently synchronized.

3. Verify the data on the data source using the DB2 Command Line Processor:

- a. Open the DB2 CLP on the source database.
- b. Enter an SQL statement to select all the records of the subscribed table.

Note: If you are using the standalone version of DB2 Version 9.1, you can also verify the data on the source database using the DB2 Version 9.1 Control Center. Right-click the subscribed table and select **Sample Contents** to browse the contents of the table.

4. Compare the contents of the table on the mobile device to that of the data source.

Related concepts

“The DB2 Sync sample application” on page 138

“DB2 Sync menu options” on page 139

This topic describes the menu options for configuring DB2 Sync.

Related tasks

“Configuring Server settings on DB2 Sync for a Palm OS device” on page 139

This topic describes how to configure the Server settings field that is necessary for successful synchronization.

Enabling and disabling synchronization

Before synchronization can happen, both the user and group must be enabled in the Mobile Devices Administration Center. If you delete required information from an object at a later time, the object is automatically disabled.

Prerequisite:

Create the user in the Mobile Devices Administration Center before his or her device is registered during the first synchronization. See “Allowing multiple devices per user” on page 21 for information about letting one user synchronize multiple mobile devices.

- You can enable a group for synchronization after you assign to the group:
 - A unique name
 - At least one user
 - At least one subscription set
- A user is automatically enabled for synchronization after you:
 1. you provide a unique name for the user.
 2. you assign the user to an enabled group.
 3. the user’s device is registered.

Related tasks

“Enabling synchronization” on page 26

This topic introduces two approaches you can use to enable synchronization.

“Disabling synchronization” on page 19

You might choose to disable an object temporarily if you are making administration changes to either the group or the user. This topic introduces how to disable

DB2 Everyplace support and troubleshooting

This topic presents tools, utilities, and techniques that you can use to identify and solve DB2 Everyplace problems.

Testing the DB2 Everyplace Sync Server servlet

Follow these steps to make sure that the DB2 Everyplace Sync Server is working properly.

Prerequisite: To ensure that the DB2 Everyplace Sync Server can successfully communicate with clients, make sure that the servlet is running. You do not need to have any clients installed in order to perform a check.

1. Start the DB2 Everyplace Sync Server.
2. In a Web browser, go to `http://yourhostname.yourdomain:8080/db2e/db2erdb`. 8080 is the default HTTP port number. Substitute port 8080 with the correct port number of the DB2 Everyplace Sync Server, which is listed in the Post-install summary panel which is displayed at the end of the DB2 Everyplace installation. If the servlet is running correctly, it will display the following message: DB2e SyncServer <datetime string>
3. For the Application Server - Express that ships with DB2 Everyplace, the server log files located in `$DSYINSTDIR/Server/logs/servername` might contain a message saying that the port is already taken by another program. If you receive this message, change the port number. The default value during installation for the server name is IBMDB2eServer. The default value during installation for the server IP address is 127.0.0.1.
4. **Optional:** To change the port number:

Windows systems

Issue the following commands:

```
cd [DSYINSTDIR]\Server\installableApps\common [DSYINSTDIR]\WAS\bin\stopServer.bat
IBMDB2eServer [DSYINSTDIR]\WAS\bin\wsadmin.bat -conntype NONE -f
dsyConfigEmbedded.jacl DefaultNode default_host DB2eServer 127.0.0.1 [DSYINSTDIR]
[new http port] [new https port]
```

Linux or UNIX systems

Issue the following commands:

```
cd [DSYINSTDIR]/Server/installableApps/common [DSYINSTDIR]/WAS/bin/stopServer.sh
IBMDB2eServer [DSYINSTDIR]/WAS/bin/wsadmin.sh -conntype NONE -f
dsyConfigEmbedded.jacl DefaultNode default_host DB2eServer 127.0.0.1 [DSYINSTDIR] [new
http port] [new https port]
```

Related tasks

“Starting the application server” on page 147

Diagnostic data for the DB2 Everyplace mobile database

DB2 Everyplace provides the following methods to log the activity of your applications. This data can help you troubleshoot problems throughout the development and testing process.

Trace files

Tracing allows you to view detailed information about each transaction between your application and the DB2 Everyplace database engine. To enable tracing, install the development libraries on the mobile device.

Log files

When an application encounters a severe system error (SQLState 58005), DB2 Everyplace logs the error in a log file.

Dump files

When an application encounters a severe system error (SQLState 58005), DB2 Everyplace captures the system state in a dump file.

The following table lists the names of each type of diagnostic file. DB2 Everyplace creates these files in the same directory as the mobile database.

Table 8. Tracing and diagnostics file names

| Type of | Long file name | Short file name (8.3 format) |
|------------|----------------|------------------------------|
| Trace file | DSY_DB2eTRACE | _trc.DBs |
| Log file | db2ediag.log | _diag.DBs |
| Dump files | DB2e_cxxxxx | _cxxxx.DBs |

where *xxxxx* is the process identifier (PID) number of the affected process.

Diagnostic data for the DB2 Everyplace Sync Server

Use the following files to troubleshoot problems with the DB2 Everyplace Sync Server or the Mobile Devices Administration Center.

Table 9. Log and trace files for the DB2 Everyplace Sync Server and Mobile Devices Administration Center

| File type | Path |
|---|---|
| DB2 Everyplace Sync Server log file | <DSYPATH>\Server\logs\IBMDB2eServer\syncadapterinit.log |
| DB2 Everyplace Sync Server trace file | <DSYPATH>\Server\logs\IBMDB2eServer\dsynmmmm.trace |
| Mobile Devices Administration Center trace file | <DSYPATH>\Server\logs\dsyadminmmmm.trace |

Important: If you are using Windows, view the log and trace files in Wordpad. Non-English characters in the log files might not display correctly if you view them from the command prompt.

To change the directory to which DB2 Everyplace stores its log files, edit the DSYGdflt.properties file, which is located in the <DSYPATH>\Server\properties\com\ibm\mobileservices directory. Set the Trace.Path property to the directory to which you want the log files to be stored. Use "\\\" to denote subdirectories.

Example: To store the log files in the C:\log\DB2e directory, enter the following value for the Trace.Path property:c:\\log\\DB2e

Enabling tracing for the DB2 Everyplace Sync Client

Each time that you synchronize a mobile device, the DB2 Everyplace Sync Client generates a trace file called trace-isyn that stores information about the synchronization process. Follow these steps to enable tracing in your CLI-based application.

1. Call the `iscServiceOpenEx()` function to create a new service handle. When you call the function, set the value of the `isync.trace` argument to "detailed".
2. Call the `iscConfigOpen()` function to connect to the configuration store. When you call the function, set the value of the `path` argument to the path to which you want to store the trace-isyn file.

Important: For debugging purposes, view the file by using a text editor on a workstation. If you are using Windows, open the trace file in Wordpad. Windows Notepad might not display international characters properly.

Important: When you report a synchronization problem, include a detailed-level trace file.

Verifying database integrity with the data integrity check tool

The data integrity check tool reports whether tables and indexes are corrupted.

To run this tool, execute the DBCHECK command in DB2eCLP. The command syntax is:

```
DBCHECK outputfile
```

The *outputfile* parameter specifies a text file in the database directory where the tool will write the results.

Note: This tool is supported on Linux™ and Windows 32-bit operating systems only.

Handling DB2 Everyplace synchronization problems

When synchronization is interrupted, the DB2 Everyplace Sync Server writes messages to the log in the administration control database.

The following topics explain how to handle synchronization problems:

Synchronization conflict resolution

At times, changes that a client submits to the DB2 Everyplace Sync Server conflict with changes that other clients or applications previously made or are simultaneously making to the source tables. The DB2 Everyplace Sync Server tracks the version of each record in each table in a replication subscription. Each client is similarly tracked to maintain a version of each record for each client's last synchronization with each table. This information allows the DB2 Everyplace Sync Server to determine whether a client is attempting to update a row based on an obsolete version of the data for that row. If a client attempts to update a row based on an obsolete version of the data for that row, the update is rejected.

Note: Restoring a DB2 Everyplace target (device) database from an earlier backup image will result in unpredictable behavior, including data inconsistencies that might affect users in addition to the user of the offending device.

Conflict resolution happens when data is staged to the mirror tables on the mid-tier system, as shown in Figure 9 on page 158. This occurs in the replication cycle following the client's synchronization session. As a result, conflicts from a client's updates will not be detected until after response messages are returned to the client during that synchronization. Rejections of client changes will be communicated back to the client in the first synchronization session following the replication in which the conflict was discovered. If a client change is based on an obsolete record, a correct version of that record will be returned in the original synchronization request.

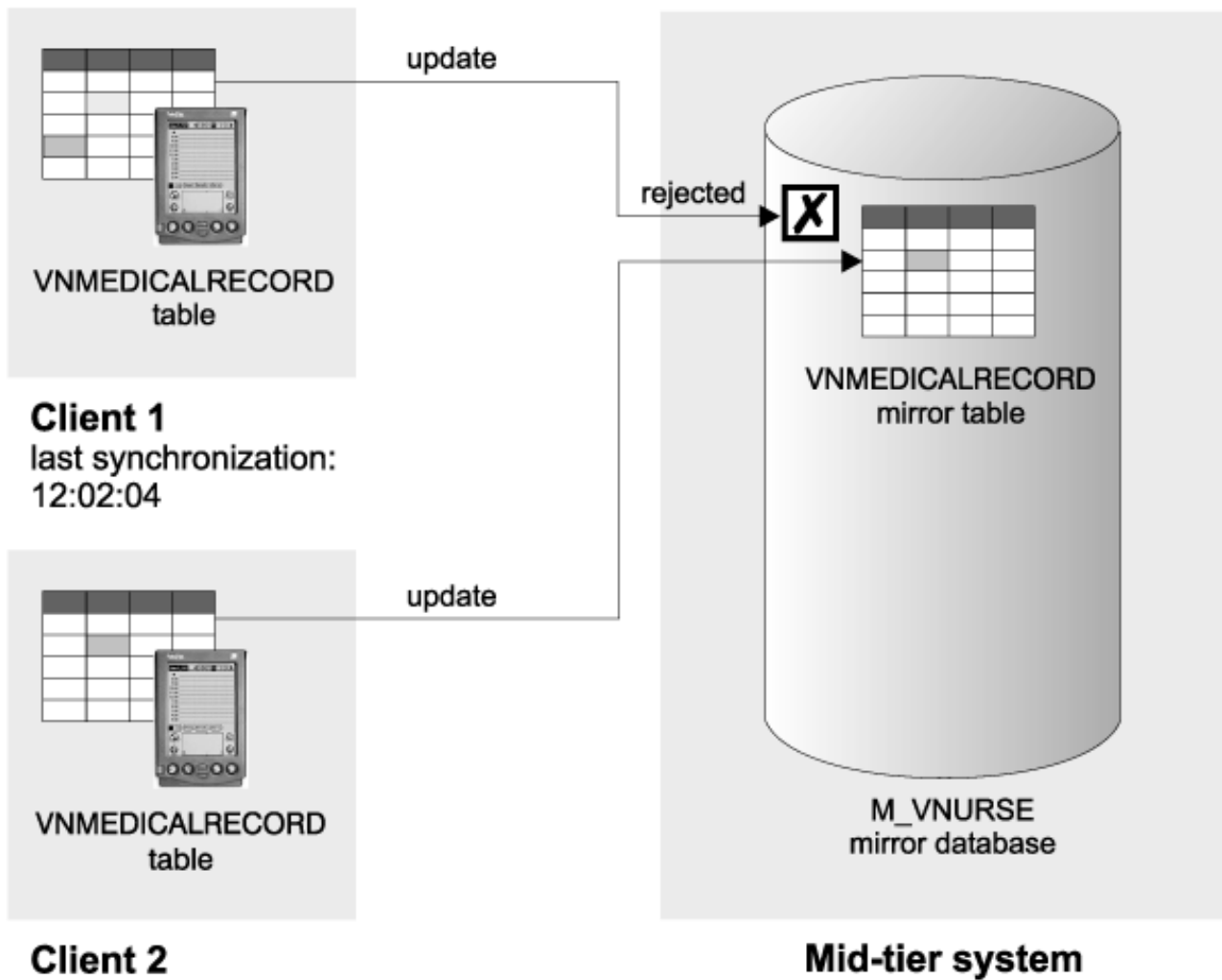


Figure 9. How the DB2 Everyplace Sync Server handles conflicts

The client whose update was rejected receives both the rejected record and a correct version of that record. The rejected record is recorded in the log on the client or provided to the application by the client API. The correct version of that record replaces the original (rejected) record on the client's DB2 Everyplace mobile database.

When DataPropagator applies the changed data from the mid-tier to the source database, additional types of conflicts can occur. See the DB2 Version 9.1 documentation for more information about how these conflicts are managed and resolved.

Related concepts

“Synchronization conflict resolution” on page 157

“DB2 Everyplace environments” on page 5

DB2 Everyplace Enterprise Edition is a robust solution for synchronizing enterprise data. You can configure the DB2 Everyplace environment in multiple ways depending on the needs of your network and your users.

The order of synchronization and reception of error messages

Subscriptions are synchronized in the order that you added them when you created the subscription set in the Mobile Devices Administration Center. Similarly, for each subscription, the tables are synchronized in the order that you added them when you created or altered the subscription. It's important to understand the order of synchronization so that you can interpret the logs and resolve synchronization problems.

You can change the order of the subscriptions and subscription sets by editing them in Mobile Devices Administration Center. The subscriptions listed in the Change Subscription Set or Change Subscription window are in the order that you added them, with the earliest at the top. You can alter the order of synchronization in the Change Subscription Set, Create Subscription Set, Change Group, Create Group, and Define Replication Subscriptions windows.

If a record was rejected by the DB2 Everyplace Sync Server for JDBC and DataPropagator subscriptions, the client receives error messages about the rejection during the synchronization that follows a replication of that mirror database.

Related concepts

“Handling DB2 Everyplace synchronization problems” on page 157

When synchronization is interrupted, the DB2 Everyplace Sync Server writes messages to the log in the administration control database.

Related tasks

“Viewing the error log to diagnose problems”

“Purging error log entries automatically” on page 161

“Defining the tracing level” on page 160

“Providing error-handling logic for user-exits” on page 161

“Viewing the log on the mobile device” on page 161

Viewing the error log to diagnose problems

When you encounter synchronization problems, you can troubleshoot by using the Mobile Devices Administration Center to view the error log.

To determine the actions that you should take for a particular message, see “Error messages” on page 197, which explains each message code and suggests actions that you can take to resolve the problem.

Additional logs are created in the form of trace files. You can use a text editor to view the trace files. The location of the trace files are defined by the Trace.Path value which is found in the **DSYGdflt.properties** file. The value in Trace.Path is identified by the .trace suffix. By default, this value is set to store the trace files in the <DSYPATH>\Server\logs directory, where <DSYPATH> is the directory where DB2 Everyplace is installed. You can change the default value by editing the Trace.Path value.

For Websphere Everyplace Access customers, the location of the trace files (for the DB2 Everyplace Sync Server Servlet) are found in the WAS node for the DB2 Everyplace Sync Server Servlet. By default the location of the trace files are set to the <DSYPATH>\Server\logs directory, where <DSYPATH> is the directory where DB2 Everyplace is installed. This value can be changed, so it's advisable to check for the current location of the trace files. The trace file location for errors in the Mobile Devices Administration Center, replication involved inside by the DB2 Everyplace Sync Server servlet, and the XML Scripting tool are located the DB2 Everyplace **DSYGdflt.properties** file.

To view the log to diagnose problems:

1. Start the Mobile Devices Administration Center.
2. In the object tree, select the Logs folder to open the log.

When you open the **Logs** folder, the contents pane displays the following information:

Timestamp

This field displays the time the message was written to the log.

Code This field displays the number of the message.

Description

This field displays the text of the message. Message text is truncated to 255 characters.

User name

User for which this log entry is associated, if applicable.

Subscription

Subscription for which this log entry is associated, if applicable.

Database

Database for which this log entry is associated, if applicable.

Host:Port

Identification of DB2 Everyplace Sync Server reporting this log entry.

Defining the tracing level

By default, the DB2 Everyplace Sync Server only logs error messages in the trace files. However, for diagnostic purposes, you might want to turn on tracing to include more detailed information. Use `DSYTrace` to turn on or off tracing.

The `DSYTrace` command changes the `Trace.Level` in file `DSYGdflt.properties` in the `\<DSYPATH>\Server\properties\com\ibm\mobileservices\` directory, where `<DSYPATH>` is the directory where DB2 Everyplace is installed. For additional tracing options, type `DSYTrace` with no options at the command line or open the `DSYGdflt.properties` file in a text editor and modify the `Trace.var` parameters where `var` is a specific parameter variable.

1. To save all trace messages in a `.trace` file, turn tracing on.
 - a. Open a command prompt.
 - b. Change to the `<DSYPATH>/Server/bin` directory, where `<DSYPATH>` is the directory where DB2 Everyplace is installed.
 - c. At the command prompt, type: `DSYTrace 1 -console`
2. To save only error messages in a `.trace` file, turn tracing off.
 - a. Open a command prompt.
 - b. Change to the `<DSYPATH>/Server/bin` directory, where `<DSYPATH>` is the directory where DB2 Everyplace is installed.
 - c. At the command prompt, type: `DSYTrace 0 -console`

Related concepts

“Handling DB2 Everyplace synchronization problems” on page 157

When synchronization is interrupted, the DB2 Everyplace Sync Server writes messages to the log in the administration control database.

“The order of synchronization and reception of error messages” on page 159

Related tasks

“Viewing the error log to diagnose problems” on page 159

“Purging error log entries automatically” on page 161

“Providing error-handling logic for user-exits” on page 161

“Viewing the log on the mobile device” on page 161

Viewing the log on the mobile device

If you are using DB2 Sync on a Palm device or emulator, a synchronization log is kept (LOGDB-ISYN.pdb). To view the log in the DB2 Sync application, click the **Log** button. If you want to save its contents for debugging purposes, use any utility that can view the contents of a .pdb file to open this file.

The contents of the synchronization log are overwritten each time a new synchronization starts.

In addition to the synchronization log that the DB2 Sync provides, the synchronization engine generates a trace file named trace-isyn.

Related concepts

“Handling DB2 Everyplace synchronization problems” on page 157

When synchronization is interrupted, the DB2 Everyplace Sync Server writes messages to the log in the administration control database.

“The order of synchronization and reception of error messages” on page 159

Related tasks

“Viewing the error log to diagnose problems” on page 159

“Purging error log entries automatically”

“Defining the tracing level” on page 160

“Providing error-handling logic for user-exits”

Purging error log entries automatically

You can specify how many days the entries in the error log in the Mobile Devices Administration Center should be kept. The log entries that are older than the specified days are automatically purged. This feature helps to keep the size of the error log small.

To purge error log entries automatically:

1. Use the dsysetproperty tool to set the DSYGdflt Log.KeepDays property.
2. You can also set a maximum number of log entries which will be stored before purging will take place. This is dictated by the DSYGdflt Log.PruneToSize property, which defaults to 10000 entries.

Related concepts

“Handling DB2 Everyplace synchronization problems” on page 157

When synchronization is interrupted, the DB2 Everyplace Sync Server writes messages to the log in the administration control database.

“The order of synchronization and reception of error messages” on page 159

Related tasks

“Viewing the error log to diagnose problems” on page 159

“Defining the tracing level” on page 160

“Providing error-handling logic for user-exits”

“Viewing the log on the mobile device”

Providing error-handling logic for user-exits

Frequently, error messages from the Mobile Devices Administration Center logs require that you take action to resolve the problem indicated in the message text. To simplify day-to-day synchronization management, you might choose to add your own logic to automatically perform these actions when a particular error message is issued.

For example, suppose that you want to include a user exit that pages you when message DSYD000E is issued. You could write a program called pager.exe that dials your pager, and includes the pager number as a parameter. The line in the DSYUserExits file might look something like this:

```
DSYD000E=pager.exe number=9980674
```

You can also include the following parameters and variables with the command:

- ID** Use this parameter to write the number of the message. For the parameter value, specify the variable *DSYID*.
- MSG** Use this parameter to write the message text. For the parameter value, specify DSYMSG to write the actual message text, or DSYMSG_ to write the message text, but convert all blank spaces in the text to underscore characters.

The following example dials the same pager, but submits both the message number and its text as parameters so they appear in the pager window:

```
DSYD000E=pager.exe number=9980674 id=dsyid msg=dsymsg
```

The DB2 Everyplace Sync Server does not check the validity of your entries in the DSYUserExits properties file, nor does it validate that the action associated with the message has been completed.

Use the DSYUserExitsTest.bat file tool to test a user exit routine. Use the following format to test:

```
DSYUserExitsTest.bat dsy_message_id
```

where *dsy_message_id* is the message number you want to simulate. If you do not provide a message ID, a list of available message IDs is displayed.

Example command with message id DSYS001I:

```
DSYUserExitsTest.bat dsys001i
```

The DSYS001I message is generated and the user exit defined for this message number is started.

Example command with an invalid message number:

```
DSYUserExitsTest.bat zzz
```

The output from the DSYUserExitsTest tool is:

```
DSYUserExitsTest
  DSY message id 'ZZZ' not found. Valid DSY message id's are:
DSYA000E, DSYA001E, DSYD000E, DSYD002E, DSYD006E, DSYD007E, ...
```

To define automatic processing of errors:

1. Open the DSYUserExits.properties file for editing. This file associates a message number with a routine or program that runs when the message number is written to the log. This properties file supports many different parameters.

```
# Formats:
# {DSY message id}={class to execute} {environment parameters}
# {DSY message id}={command to execute} {environment parameters}
#   where
#     {DSY message id}:
#       a DSY* message id that you want to define a user exit for (such as
#       DSYD000E)
#     {class to execute}:
#       the name the Java .class to execute. This class must implement the
#       com.ibm.mobileservices.DSYUserExitsInterface
#     {command to execute}:
#       the name the command to execute (such as pager.exe)
#     {environment parameters}:
#       a series of parameters to pass in to the class or command to execute
```

```

#
# Optional command tags:
# <DSYID>           = the message id
# <DSYIDMSG>        = the message id message text
# <DSYIDMSG_>       = the message id message text, but all blanks are
#                   converted to underscores
# <DSYMSG>          = the message text
# <DSYMSG_>         = the message text, but all blanks are converted to
#                   underscores.
# <SERVER_IPADDRESS> = the server ip address (such as 9.112.19.143)
# <SERVER_NAME>     = the server name (such as mpauser.stl.ibm.com)
#
# Refer to the messagestopic for available DSY message ids.
#
# Example 1:
# If you wanted to have user exists for DSYD000E, you would add a line
# similar to the following:
#
# DSYD000E=pager.exe number=5551234 id=<DSYID> msg=<DSYMSG_>
#
# When a DSYD000E message was issued, the pager.exe command would be executed
# with two environment parameters would be set: number=5551234, id=DSYD000E
# and msg=DSY message text, substituting an underscore (_) for blanks.
#
# Example 2:
# If you wanted to have a class executed when a DSYD020E was encountered,
# you would add a line similar to the following:
#
# DSYD020E=com.ibm.mobileservices.DSYUserExitsSample.class
#
# When a DSYD020E was issued, the com.ibm.mobileservices.DSYUserExitsSample
# class would be executed.

```

The action to be performed must be a reference to a valid routine or program.

- At the end of the file, pair a message number with an action to be performed when that message is written to the log. The action can be a command (such as an executable or batch file) or a Java class. Use the following format:

```
message_number=action parameter=value
```

where:

message_number

The number of the message. See “Error messages” on page 197 for message numbers and their associated text.

action The file referencing the command or Java class that is called. The Java class must implement the com.ibm.mobileservices.DSYUserExitsInterface.

parameter=value

A series of parameter sets (such as lastname=Doe firstname=John), with each set separated by a space.

Related concepts

“Handling DB2 Everyplace synchronization problems” on page 157

When synchronization is interrupted, the DB2 Everyplace Sync Server writes messages to the log in the administration control database.

“The order of synchronization and reception of error messages” on page 159

Related tasks

“Viewing the error log to diagnose problems” on page 159

“Purging error log entries automatically” on page 161

“Defining the tracing level” on page 160

“Viewing the log on the mobile device” on page 161

Monitoring Synchronization in DB2 Everyplace

Topics in this section describe how to monitor synchronization between mobile devices and the DB2 Everyplace synchronization server.

Monitoring DB2 Everyplace mobile device synchronization

You can use the tools, utilities, and features in this topic to get information about client synchronization activity.

Using them, you can generate reports that can help you monitor and analyze the operation and performance of client synchronizations.

Using the diagnosis utility

The diagnosis utility recovers data from the message store. The utility tool checks the synchronization status of each subscription that the user is associated with and checks if it needs recovery. If it does, the tool recovers and writes the recovery status in an output file. You can run this utility before deleting or resetting a device or user ID to collect data that can help you diagnose client synchronization problems.

To use the diagnosis utility:

1. Open a command window on a system that is running the DB2 Everyplace Sync Server.
2. Issue the DYSDiagnose command. The command has the following argument list:

user Specifies the user.

-file Specifies a file to store the output generated by the tool. If no file name is specified, the utility writes output to the system console.

The following command invokes DSYDiagnose specifying user nurse1 and output file c:\diagnosis_report\nurse1_report.txt.

```
DSYDiagnose nurse1 -fc:\diagnosis_report\nurse1_report.txt
```

Using the recovery utility

For the DB2 Everyplace Sync Server, a *task* is one of two things:

- Preparing a message to send to a mobile device.
- Saving data from a mobile device into the database

When a client synchronization attempt fails, you can use the recovery utility to redo DB2 Everyplace Sync Server tasks by rebuilding messages or saving changes to the database. The recovery utility and the DB2 Everyplace Sync Server can run at the same time without causing conflicts.

To use the recovery utility:

1. Open a command window on a computer that is running the DB2 Everyplace Sync Server. The command is installed by default in C:\Program Files\IBM\DB2Everyplace\Server\bin (Windows).
2. Issue the DYSRecovery command. The command takes one argument:

mirror_db
Specifies the mirror database.

The following command invokes DSYRecovery specifying M_VN2 as the mirror database. This example sends results to the console.

```
DSYRecovery M_VN2
```

```
Recovery task for Mirror DB 'M_VN2' is successfully executed.
```

Collecting synchronization information

The DB2 Everyplace Sync Client can collect information about synchronization sessions, including statistics, status, and trace data, and then upload it to the server through a regular upload subscription. The subscription set DSYCLIENTSTAT_SET, which includes the upload subscription DSYCLIENTSTAT, is created automatically when you install DB2 Everyplace. After installation, you can associate the subscription set with specific groups or users, enabling them to use this feature.

This feature collects a wide spectrum of synchronization information, including the time when synchronization starts, how long it takes, what subscriptions get synchronized, their status, the size of the transmitted data changes, the time for processing the data, and the client traces. This feature collects much more data than the DSY.LOG table, which only records the synchronization start time and whether synchronization was completed. Also, logs in the DSY.LOG table will be deprecated in future releases.

Enabling the client synchronization statistics subscription

The subscription set DSYCLIENTSTAT_SET includes the upload subscription DSYCLIENTSTAT that allows client devices to send synchronization statistics to the DB2 Everyplace Sync Server. By associating that subscription set with a group, you enable users in that group to use the client synchronization information upload feature.

To enable the client synchronization statistics subscription:

1. Start the Mobile Devices Administration Center.
2. Click **Groups** in the object tree.
3. Right-click the group you want to change, then choose **Edit** from the context menu.
4. Select **DSYCLIENTSTAT_SET** from the **Available subscription sets** list.
5. Click the > button to add DSYCLIENTSTAT_SET to the **Selected subscription sets** list.
6. Select the **Enable synchronization** check box.
7. Click **OK**.

Enabling users to upload synchronization trace data

After enabling a group of users to upload the client synchronization information, you can configure the group to also upload the client synchronization trace, which is turned off by default. The client trace upload is controlled by pre-defined data filter properties, and they can be changed by using either the Mobile Devices Administration Center or the XML scripting tool.

Setting synchronization statistics properties using the Mobile Devices Administration Center:

To configure when the client collects and uploads synchronization trace data, set data filter properties in the Group panel of the Mobile Devices Administration Center.

Important: The same properties can also be set in the User panel, which will overwrite the group settings.

To set synchronization statistics properties using the Mobile Devices Administration Center:

Perform the following steps in the Mobile Devices Administration Center.

1. Start the Mobile Devices Administration Center.
2. Click **Groups** in the object tree.
3. Right-click the group you want to change, then choose **Edit** from the context menu.
4. Click the **Data filter** tab in the Change Group window.
5. Click an item in the **Parameter name** column.
6. Click **Change**.
7. Enter a new value in the **Change Data Filter** window.
8. Click **OK** to close the **Change Data Filter** window.
9. Click **OK**.

The following properties determine when the client should collect and upload the synchronization trace data.

isync.info.limit.count

Specifies the number of rows of client information (including both statistics and trace) to collect before uploading data to the server. When the limit is reached, the client stops collecting data until the existing data is uploaded. The default limit is 5000 rows.

isync.trace.collect.status

Specifies whether the client should save the client trace into the Client Control Database or simply discard it. The supported values for this property are:

- none - do not collect the synchronization trace (default)
- error - collect the synchronization trace only if there is an error
- all - always collect the trace.

isync.trace.upload

Specifies whether the client should upload the client trace from the Client Control Database into the Upload database. The supported values for this property are "yes" and "no." "No" is the default value.

Note: If the collected information reaches the limit (set by the `isync.info.limit.count` property), the client stops collecting new synchronization information. If `isync.trace.upload` is set to "no", then new information is never collected.

Configuring synchronization information databases on the client

To support uploading of synchronization information, the client uses two logical databases:

- Client control database: collects synchronization information.
- Default target database: uploads synchronization information.

The default target database and the client control database can reside in the same physical database. An application can configure these databases by defining the `isync.db.*` and `target.db.*` properties when opening a service handle.

Configuring client synchronization information databases using the Java API:

The DB2 Everyplace Sync Client stores synchronization statistics in two databases: the client control database stores data collected during synchronization, and the default target database stores data collected for upload. An application can specify characteristics of those databases by setting properties when opening the service handle.

To configure client synchronization information databases using the Java API:

1. Call `createSyncService`.
2. Set values for the following properties.

isync.db.driver
JDBC driver for the Client Control Database

isync.db.url
URL (path) of the Client Control Database

isync.db.username
User name for the Client Control Database

isync.db.password
Password for the Client Control Database

target.db.driver
JDBC driver for the default target database

target.db.url
URL (path) of the default target database

target.db.username
User name for the default target database

target.db.password
Password for the default target database

Configuring client synchronization information databases using the C API:

The DB2 Everyplace Sync Client stores synchronization statistics in two databases: the client control database stores data collected during synchronization, and the default target database stores data collected for upload. An application can specify characteristics of those databases by setting properties when opening the service handle.

To configure client synchronization information databases using the C API:

1. Call `iscServiceOpenEx`.
2. Set values for the following properties.

isync.db.driver
JDBC driver for the Client Control Database

isync.db.url
URL (path) of the Client Control Database

isync.db.username
User name for the Client Control Database

isync.db.password
Password for the Client Control Database

target.db.driver
JDBC driver for the default target database

target.db.url
URL (path) of the default target database

target.db.username
User name for the default target database

target.db.password
Password for the default target database

Querying client synchronization information tables

The uploaded client synchronization information is stored in three tables in the DSYCSTAT database: DSY.SYNC_STATS, DSY.SYNC_SUBS_STATS, and DSY.SYNC_TRACE. Their table structures are available in Client synchronization statistics table structures.

To query the synchronization statistics tables:

The following example returns the number and status of synchronization operations performed by a specific user ('nurse1') during a specific period of time.

```
SELECT SyncOpID, ServerTS, Status
FROM DSY.SYNC_STATS
WHERE UserID = 'nurse1' AND
      ServerTS >= '2004-01-01-00.00.00' AND
      ServerTS <= '2004-12-31-23.59.59'
```

The following example finds out how many subscriptions a user ('nurse1') has successfully synchronized.

```
SELECT count(*)
FROM DSY.SYNC_SUBS_STATS
WHERE UserID = 'nurse1' AND Status = 'S'
```

The following example fetches the client trace (if turned on from Mobile Devices Administration Center) for a specific sync operation (SyncOpID = 10) that a user ('nurse1') performed.

```
SELECT LogTime, DSYCode, Msg
FROM DSY.SYNC_TRACE
WHERE SyncOpID = 10 AND UserID = 'nurse1'
ORDER BY LineNo
```

Related reference

“Client synchronization statistics table structures”

The uploaded client synchronization information is stored in three tables in the DSYCSTAT database: DSY.SYNC_STATS, DSY.SYNC_SUBS_STATS, and DSY.SYNC_TRACE.

Using the sample synchronization statistics report

Prerequisites:

- You must enable the client synchronization statistics subscription for specific groups and collect synchronization data before using the sample report.
- Required software: an application (such as Microsoft Excel) that can read XSL spreadsheet files and execute spreadsheet macros.

To use the sample synchronization statistics report:

1. Open Sample_Sync_Report.xsl. In a typical DB2 Everyplace installation, this file is installed in C:\Program Files\IBM\DB2Everyplace\Server\syncreport (Windows).
2. Click the text **Need help? Click here.** to display instructions for entering parameters, retrieving data, and generating reports.

Client synchronization statistics table structures

The uploaded client synchronization information is stored in three tables in the DSYCSTAT database: DSY.SYNC_STATS, DSY.SYNC_SUBS_STATS, and DSY.SYNC_TRACE.

This topic presents the structures of the client synchronization statistics tables in the DSYCSTAT database:

- DSY.SYNC_STATS
- DSY.SYNC_SUBS_STATS
- DSY.SYNC_TRACE

Table 10. DSY.SYNC_STATS table structure

| Column Name | Column Type | Description |
|-------------|--------------|--|
| UserID | VARCHAR(254) | DB2 Everyplace Sync Client user name. |
| DeviceID | VARCHAR(56) | Device ID. |
| SyncOpID | INT | Sync operation ID. |
| ClientTS | TIMESTAMP | Initial timestamp (in the client local time zone) on the device. |
| ServerTS | TIMESTAMP | Timestamp in GMT (Greenwich Mean Tme) on the server when it replied the first client request. |
| SyncTime | INT | Total synchronization time (in milliseconds). |
| Status | CHAR(1) | Synchronization status: <ul style="list-style-type: none"> • 'S' : Succeeded • 'F' : Failed • 'C' : Canceled • 'X' : Disabled • 'N' : No change |

Table 11. DSY.SYNC_SUBS_STATS table structure

| Column Name | Column Type | Description |
|-------------|--------------|--|
| UserID | VARCHAR(254) | DB2 Everyplace Sync Client user name. |
| DeviceID | VARCHAR(56) | Device ID. |
| SyncOpID | INT | Sync operation ID. |
| SessionID | INT | Session ID. |
| SubsID | VARCHAR(30) | Subscription ID. |
| SubsName | VARCHAR(20) | Subscription name. |
| ClientTS | TIMESTAMP | Initial timestamp (in the client local time zone) on the device for synchronizing the subscription. |
| ServerTS | TIMESTAMP | Timestamp on the server in GMT (Greenwich Mean Tme) when it replied to the first client request for synchronizing the subscription. |
| SyncMode | CHAR(1) | Synchronization mode: <ul style="list-style-type: none"> • 'R' : Refresh • 'S' : Synchronize |
| Resume | CHAR(1) | Is the session resuming a previously cancelled session? <ul style="list-style-type: none"> • 'T' : True (yes) • 'F' : False (no) |
| Status | CHAR(1) | Synchronization status: <ul style="list-style-type: none"> • 'S' : Succeeded • 'F' : Failed • 'C' : Canceled |

Table 11. DSY.SYNC_SUBS_STATS table structure (continued)

| Column Name | Column Type | Description |
|----------------|-------------|--|
| DSYCError | INT | DSYC error code if synchronization fails, or 0 if synchronization succeeds. |
| TotalRequests | INT | Total number of requests the client sent to the server during synchronization. |
| FailedRequests | INT | Total number of requests where the client did not receive the requested data. |
| BytesSent | INT | Total number of bytes sent from client to the server during synchronization. |
| BytesRecv | INT | Total number of bytes that the client received from the server during synchronization. |
| SendTime | INT | Total time (in milliseconds) the client takes (in network transportation) to send the client data to the server. |
| RecvTime | INT | Total time (in milliseconds) the client waits to receive the server data. |
| BuildTime | INT | Total time (in milliseconds) the client takes to fetch the client changes and build a message to send to the server. |
| ApplyTime | INT | Total time (in milliseconds) the client takes to apply the operations in the server messages. |
| SyncTime | INT | Total synchronization time for the session (in milliseconds). |
| UnitsUploaded | INT | Total number of logical units of the local changes that the client uploads to the server. Its semantics varies by the subscription type: <p>Config subscription</p> <ul style="list-style-type: none"> • 0 : no config change • 1 : any config change <p>Table subscription (JDBC or Upload)</p> <p>The number of records.</p> <p>File subscription</p> <p>The actual file size (in bytes).</p> |

Table 11. DSY.SYNC_SUBS_STATS table structure (continued)

| Column Name | Column Type | Description |
|--------------|-------------|--|
| UnitsApplied | INT | Total number of logical units of the server changes that the client applies to the target data.

Config subscription <ul style="list-style-type: none"> • 0 : no config change • 1 : any config change Table subscription (JDBC or Upload)
The number of records.

File subscription
The actual file size (in bytes). |

Table 12. DSY.SYNC_TRACE table structure

| Column Name | Column Type | Description |
|-------------|--------------|---|
| UserID | VARCHAR(254) | DB2 Everyplace Sync Client user name. |
| DeviceID | VARCHAR(56) | Device ID. |
| SyncOpID | INT | Sync operation ID. |
| LineNo | INT | Line number of the trace. |
| LogTime | TIMESTAMP | Timestamp (in the client local time zone) of the log entry. |
| DSYCode | CHAR(9) | DSY Error Code. |
| Msg | VARCHAR(255) | Trace message. |

Monitoring DB2 Everyplace server synchronization

This section explains the different options for server synchronization in DB2 Everyplace.

Specifying pruning options for the message store

The DB2 Everyplace Sync Server archives each message exchanged between the server and the client in the MSGDUMPS folder (for file system message store users) or the DSY.MSGSTORE table (for database message store users) for debugging purposes. Because the message store can take up a lot of space over time, the DB2 Everyplace Sync Server executes a task every hour (3600 seconds) to prune the message dumps. By default, only messages older than 7 days are pruned. You can set the frequency and amount to prune using the dsysetproperty.

To specify pruning options for the message store:

- If you want to set the frequency of pruning, run `dsysetproperty DSYScheduledTasks com.ibm.db2ess.messagestores.utils.MsgDumpsPruneTask=time_in_seconds` where *time_in_seconds* is the frequency of pruning in seconds.
- If you want to set the amount of days the message store keeps a message, run `dsysetproperty DSYIdflt messagestore.msgdumps.prune.days=days` where *days* is the amount of days to keep a message in the message store.
- To validate or view the current value of these properties, run

dsysetproperty DSYScheduledTasks com.ibm.db2ess.messagestores.utils.MsgDumpsPruneTask to view the value for the frequency of pruning, or run

dsysetproperty DSYIdflt messagestore.msgdumps.prune.daysto view the value for the amount of days to keep messages in the message store.

DB2 Everyplace Sync Server replication statistics

The DB2 Everyplace Sync Server collects statistics about how data is replicated to mobile devices. You can use this information to debug your applications and analyze the performance of your DB2 Everyplace environment.

The type of statistics that the DB2 Everyplace Sync Server records depends upon the type of replication that you are using. DB2 Everyplace stores this information in the DSY.REPLSTATISTICS table within the mirror database.

JDBC replication

- Number of inserts downloaded and time taken
- Number of deletes downloaded and time taken
- Number of updates downloaded and time taken
- Number of rows pruned in the source history table and time taken
- Number of inserts uploaded and time taken
- Number of deletes uploaded and time taken
- Number of updates uploaded and time taken
- Number of mirror conflicts, master conflicts and peer-peer conflicts

DataPropagator replication

- Number of records moved from staging table to mirror and time taken
- Number of incomplete stage records moved to prior sync window and time taken.
- Number of mirror and peer conflicts
- Time taken to run local capture on mirror database
- Time taken to run bi-directional apply
- Time taken to run local capture on mirror database
- Number of records pruned in change data table and time taken
- Number of records updated in mirror history and time taken

Related tasks

“Specifying pruning options for replication statistics”

The DB2 Everyplace Sync Server collects statistics about how data is replicated to mobile devices. You can use this information to debug your applications and analyze the performance of your DB2 Everyplace environment.

Specifying pruning options for replication statistics

The DB2 Everyplace Sync Server collects statistics about how data is replicated to mobile devices. You can use this information to debug your applications and analyze the performance of your DB2 Everyplace environment.

Depending on the size of your DB2 Everyplace environment, the DB2 Everyplace Sync Server might collect a large amount of replication statistics. Every hour (3600 seconds), the DB2 Everyplace Sync Server prunes statistics that are older than 15 days.

Replication statistics can require a large amount of disk space depending on how frequently the DB2 Everyplace Sync Server prunes the statistics and how many days that the statistics are kept. The following table shows example storage requirements for the DSY.REPLSTATISTICS table.

Table 13. Growth characteristics of the DSY.REPLSTATISTICS table.

| Number of subscribed tables | Row length (bytes) | Number of rows per table | Replication frequency (minutes) | Pruning threshold (days) | Approximate statistics table size (MBs) |
|-----------------------------|--------------------|--------------------------|---------------------------------|--------------------------|---|
| 100 | 50 | 15 | 15 | 15 | 112 |
| 100 | 50 | 15 | 15 | 30 | 225 |
| 100 | 50 | 15 | 15 | 100 | 750 |
| 50 | 50 | 15 | 15 | 15 | 56 |
| 50 | 50 | 15 | 15 | 30 | 112 |
| 50 | 50 | 15 | 15 | 100 | 375 |

Important: To perform the following tasks on a Windows system, log in as the system administrator. To perform the following tasks on a Linux or UNIX system, log in as the DB2 Everyplace Sync Server instance.

- To set the frequency that the DB2 Everyplace Sync Server prunes replication statistics, issue the following command: `dsysetproperty DSYScheduledTasks com.ibm.db2ess.prune.ReplStatPruneTask=time_in_seconds` where *time_in_seconds* is the frequency of pruning in seconds.
- To set the number of days that the DB2 Everyplace Sync Server keeps replication statistics, issue the following command: `dsysetproperty DSYIdflt ReplStat.pruneAge=days` where *days* is the number of days the replication statistics records will be kept before being pruned.

Related concepts

“DB2 Everyplace Sync Server replication statistics” on page 173

The DB2 Everyplace Sync Server collects statistics about how data is replicated to mobile devices. You can use this information to debug your applications and analyze the performance of your DB2 Everyplace environment.

Reference for DB2 Everyplace

This section presents reference information.

Command-line scripts for the DB2 Everyplace Sync Server

DB2 Everyplace provides scripts to operate and to administer the DB2 Everyplace Sync Server.

The administration commands can be found in the following paths:

- On Windows, these scripts are located in <DSYPATH>/Server/bin.
- On UNIX and Linux platforms, these scripts are located in the \$HOME/db2everyplace82/Server/bin directory, where \$HOME is the home directory of the DB2 Everyplace Sync Server instance. You must run these commands from the \$HOME/db2everyplace82/Server/bin directory and the command must be preceded with "./".

Important:

1. You must be logged in as the DB2 Everyplace Sync Server instance in order to execute these scripts.
2. You must perform the post-installation configuration tasks that are specific to the operating system of your server prior to using the DB2 Everyplace Sync Server and Mobile Devices Administration Center.

Example: Export the DISPLAY variable to a valid display prior to executing these scripts.

The commands for UNIX, Linux, and Windows platforms are similar. UNIX and Linux scripts have a .sh extension, and Windows batch files have a .bat extension. Use the extension that is appropriate for your platform.

Table 14 describes the administration commands that are provided by DB2 Everyplace.

Table 14. Commands for UNIX, Linux, and Windows

| Command | Description |
|----------|--|
| dsyadmin | <p>This command starts the DB2 Everyplace Mobile Device Administration Center, which provides a GUI to administer users, groups, subscription sets, and subscriptions.</p> <p>See “What is the Mobile Devices Administration Center?” on page 9 for more information about using Mobile Device Administration Center.</p> <p>Important: For Linux servers, you must start the DB2 JDBC Applet Server before you start the Mobile Device Administration Center. To start the DB2 JDBC Applet Server, type db2jstrt. The default port for the DB2 JDBC Applet Server is 6789. You can provide a free port as an optional parameter to this script. You must restart the DB2 JDBC Applet Server after restarting your computer. If you start the DB2 JDBC Applet Server on a port other than 6789, you must provide that port to dsyadmin.</p> <p>Usage</p> <p>To start dsyadmin on the default port, type:</p> <pre>dsyadmin</pre> <p>To start dsyadmin on port 765, type:</p> <pre>dsyadmin 765</pre> |

Table 14. Commands for UNIX, Linux, and Windows (continued)

| Command | Description |
|-------------|--|
| dsyadminxml | <p>This command runs the XML scripting tool. The XML scripting tool uses XML files to administer subscription objects including users, groups, subscriptions, and subscription sets.</p> <p>See “About the XML scripting tool” on page 85 for complete usage information.</p> <p>Usage</p> <p>Command-line parameters:</p> <pre>-d <filename> {-v7} -x <filename> { -add -remove } { -include <objectList> } { -exclude <objectList> } { -detailed }</pre> <p>where</p> <p>-x Create XML file from control database.</p> <p>-d Apply XML file to control database.</p> <p><filename>
XML file name to process. The -x flag will make a backup of the file if it already exists.</p> <p>-add This option is used with the -x flag. If you specify this option, the XML scripting tool generates XML that can be used to add synchronization objects to a control database.</p> <p>-remove
This option is used with the -x flag. If you specify this option, the XML scripting tool generates XML that can be used to remove synchronization objects from a control database.</p> <p>-exclude
This option is used with the -x flag. If you specify this option, the XML scripting tool excludes the objects that you specify in <i>objectList</i>. All other objects are included.</p> <p>-include
This option is used with the -x flag. If you specify this option, the XML scripting tool includes the objects that you specify in <i>objectList</i>. All other objects are excluded.</p> <p>-v7 Avoids parser errors when you apply XML documents that were generated with the DB2 Everyplace version 7 XML scripting tool.</p> <p>The <i>objectList</i> variable used by the -exclude and -include arguments is a space-separated list of one or more of the following objects:</p> <ul style="list-style-type: none"> • users • groups • subs • subsets • mirror databases • master databases • adapters • properties <p>Example: To generate XML for subscriptions only, issue the following command:</p> <pre>dsyadminxml -x myfile.xml -include subs</pre> |

Table 14. Commands for UNIX, Linux, and Windows (continued)

| Command | Description |
|----------|---|
| dsyadvis | <p>This command recommends a configuration of the DB2 Everyplace Sync Client, DB2 Everyplace Sync Server, and databases that are used by the DB2 Everyplace Sync Server.</p> <p>Usage</p> <pre>dsyadvis <component> {<parameter> <value>}</pre> <p>Restriction: You can use only one of the three component options when you execute the command. If you want recommendations for more than one component, execute the command again.</p> <p>Syntax</p> <pre>dsyadvis -c -dev <device type> -devcpu <device cpu speed> -devmem <device memory> -devcon <device connection type or speed> -dest <destination file location> dsyadvis -s -syncsph <synchronizations per hour> -upsz <upload data size> -dnsz <download data size> -tsub <total number of subscriptions> -dest <destination file location> dsyadvis -d -o -x <xml file name> -dest <destination file location></pre> <p>where</p> <ul style="list-style-type: none"> -c -C client component. The output file for this mode is called DSY_Client.out. -s -S server component. The output file for this mode is called DSY_Server.out. -d -D database component. <p>Mobile device parameters</p> <ul style="list-style-type: none"> -dev device type. This can be: <ul style="list-style-type: none"> EMBEDLINUX a mobile device that runs embedded Linux LINUX a mobile device that runs Linux PALM a mobile device that runs Palm OS POCKETPC a Pocket PC mobile device SYMBIAN a mobile device that runs Symbian OS WIN32 a mobile device that runs Windows -devcpu CPU speed of the mobile device, in megahertz -devmem memory of the mobile device. Write this value as a number followed by KB for kilobytes or MB for megabytes. |

Table 14. Commands for UNIX, Linux, and Windows (continued)

| Command | Description |
|----------------------|--|
| dsyadvis (continued) | <p>Mobile device parameters (continued)</p> <p>-devcon
mobile device connection type or speed. You can specify the following connection types:</p> <p>BLUETOOTH
a Bluetooth connection</p> <p>DSL a DSL connection</p> <p>ETHERNET
an Ethernet connection</p> <p>GPRS a General Packet Radio Service (GPRS) connection</p> <p>GSM a Global System for Mobile Communications (GSM) connection</p> <p>MODEM9K
a 9.6 Kilobit per second modem connection</p> <p>MODEM33K
a 33.6 Kilobit per second modem connection</p> <p>MODEM56K
a 56 Kilobit per second modem connection</p> <p>WIFI-B an 802.11b wireless connection</p> <p>WIFI-G
an 802.11g wireless connection</p> <p>USB1 a Universal Serial Bus (USB) Version 1.0 or Version 1.1 connection</p> <p>USB2 a USB Version 2.0 connection</p> <p>Instead of specifying a connection type, you can specify the connection speed. If you specify an actual connection speed, include the letters Mbps for megabytes per second or Kbps for kilobytes per second.</p> <p>DB2 Everyplace server parameters</p> <p>-syncsph
synchronizations per hour. This can be any whole number.</p> <p>-upsz upload data size. This value is a number followed by KB for kilobytes or MB for megabytes.</p> <p>-dnsz download data size. This value is a number followed by KB for kilobytes or MB for megabytes.</p> <p>-tsub total number of subscriptions. This can be any whole number.</p> <p>Database parameters</p> <p>-o online mode. In this mode, dsyadvis creates two files, advis_script and DSY_DSYSYSTAT.ddl. To apply the recommended database settings, run the advis_script script.</p> <p>-x xml file mode</p> |

Table 14. Commands for UNIX, Linux, and Windows (continued)

| Command | Description |
|-----------------------|--|
| dtsyadvis (continued) | <p>Common parameters</p> <p>-dest destination file location</p> <p>Example: To request a suggested configuration for a PocketPC mobile device with the following specifications:</p> <ul style="list-style-type: none"> • 300 megahertz processor • 32 megabytes of memory • Bluetooth connection to the DB2 Everyplace Sync Server <p>issue the following command:</p> <pre>dtsyadvis -c -dev POCKETPC -devcpu 300 -devmem 32MB -devcon BLUETOOTH</pre> |
| dtsyblocksubscription | <p>This command blocks a subscription. If you block a subscription, the DB2 Everyplace Sync Server rejects new requests to synchronize the subscription.</p> <p>Usage</p> <pre>dtsyblocksubscription <name></pre> <p>where <name> is the subscription to block on the server. You can block multiple subscriptions by entering the name of each subscription separated by a space.</p> |
| dtsycheck | <p>This command checks DB2 Everyplace Sync Server for inconsistencies.</p> <p>Note: It is intended for use at the request of IBM Software Support only. Run the tool only when a DB2 Everyplace Sync Server administrator is not using the Mobile Devices Administration Center or the XML scripting tool and when a database administrator is not modifying the databases used by the DB2 Everyplace Sync Server.</p> <p>Usage</p> <pre>dtsycheck {-system -mirror <mirror_database_name> -subscription <subscription_name> -table <table_schema> <table_name>}</pre> <p>dtsycheck Prints out a help message explaining the usage of the different parameters</p> <p>-system Check the entire system for consistency errors</p> <p>-mirror Check the specified mirror database for consistency errors</p> <p>-subscription Check the specified subscription</p> <p>-table Check the specified table for consistency errors</p> |
| dtsydiagnose | <p>This command checks the synchronization status of each subscription with which the user is associated and checks if the subscription needs to be recovered. If recovery is required, the tool recovers and writes the recovery status in an output file.</p> <p>Usage</p> <pre>dtsydiagnose <username> <filename></pre> <p>where <i>username</i> is the name of the user whose subscriptions you want to check, and <i>filename</i> is the name of the file where you want the recovery status to be recorded.</p> |

Table 14. Commands for UNIX, Linux, and Windows (continued)

| Command | Description |
|--------------|--|
| dsyencrypt | <p>This command encrypts the user input and displays it to the screen.</p> <p>Usage
 dsyencrypt "<input string>"</p> <p>Example: To encrypt the word IBM, issue the following command:
 dsyencrypt "IBM"</p> <p>When you issue this command, dsyencrypt displays the following output:
 Encrypted form of your input text is: R+V1lqugq4s=
 If this is an encrypted password you want to place in a properties file, then the value you should place in the properties file is: {DSY}R+V1lqugq4s=</p> |
| dsyrecovery | <p>This command recovers unsuccessful synchronizations.</p> <p>Usage
 dsyrecovery <mirror> <subscription> <device></p> <p>where</p> <p>mirror The mirror database to recover</p> <p>subscription
 The subscription to recover</p> <p>device The device to recover</p> |
| dsyreplicate | <p>When you issue this command, DB2 Everyplace replicates the data that is located in the mirror database to the source database at the first opportunity.</p> <p>Usage
 dsyreplicate <mirror_database> {-timeout <timeout_value>}</p> <p>where</p> <p>mirror_database
 the name of the mirror database</p> <p>-timeout
 When more than one process tries to replicate data from the same mirror database, only the first request is accepted and the rest are blocked. By default, the pending will time out in 90 seconds. Yet, you can set a different value using this parameter.</p> |

Table 14. Commands for UNIX, Linux, and Windows (continued)

| Command | Description |
|----------------|---|
| dsyreset | <p>This command resets a user, mobile device, or group of users.</p> <p>Usage</p> <pre>dsyreset <user name> -device <device ID> -group <group name> -all</pre> <p>where</p> <p>user name
The user name to reset. Each of the mobile devices that is registered to the group is refreshed the next time that the mobile device is synchronized. You cannot use this parameter with any other parameter.</p> <p>device ID
The mobile device ID to reset. The mobile device is refreshed the next time that any of its users synchronize it. You cannot use this parameter with any other parameter.</p> <p>group name
The group name to reset. Each of the mobile devices that is registered to the group is refreshed the next time that the mobile device is synchronized. You cannot use this parameter with any other parameter.</p> <p>-all Reset all mobile devices, users, and groups. Each mobile device is refreshed the next time that it is synchronized. You cannot use this parameter with any other parameter.</p> |
| dsysetenv | <p>This command sets the environment variables that are specific to the DB2 Everyplace Sync Server.</p> <p>Usage</p> <pre>dsysetenv</pre> |
| dsysetjavahome | <p>This script adds JDBC drivers to the classpath for the Mobile Devices Administration Center. To add JDBC drivers, modify the script. The JDBC drivers must be added to the application server classpath that runs the DB2 Everyplace servlets.</p> <p>Usage</p> <pre>dsysetjavahome</pre> <p>Example: To add the file C:\Program Files\Acme\custom.jar to the classpath, edit the script file and change the line</p> <pre>SET JDK_EXTRA_CP=.</pre> <p>to</p> <pre>SET JDK_EXTRA_CP=.;C:\Program Files\Acme\custom.jar</pre> |

Table 14. Commands for UNIX, Linux, and Windows (continued)

| Command | Description |
|----------------|---|
| dsysetproperty | <p>This command queries or sets a property in the control database DSY.PROPERTIES table.</p> <p>Usage</p> <pre>dsysetproperty <type> ALL <type> ALL <name> type <name>=<value></pre> <p>where</p> <p>type the property type</p> <p>ALL all property types (valid for queries only)</p> <p>name the property name</p> <p>value the property value. On Linux and UNIX platforms, use a space instead of an equals sign to separate the property value from the property name.</p> <p>Example: To query all properties, issue the following command:</p> <pre>dsysetproperty ALL</pre> <p>Example: To query the property Jdbc.MaxConnections with type DSYGdf1t, issue the following command:</p> <pre>dsysetproperty DSYGdf1t Jdbc.MaxConnections</pre> <p>Example: On Windows, to set the value of the property Jdbc.MaxConnections with type DSYGdf1t to 40, issue the following command:</p> <pre>dsysetproperty DSYGdf1t Jdbc.MaxConnections=40</pre> <p>Example: On Linux and UNIX, to set the value of the property Jdbc.MaxConnections with type DSYGdf1t to 40, issue the following command:</p> <pre>dsysetproperty DSYGdf1t Jdbc.MaxConnections 40</pre> |
| dsysync | <p>This command launches the embedded version of the WebSphere Application Server Version 6 that executes the DB2 Everyplace servlets. This command takes no arguments.</p> <p>If DB2 Everyplace is installed in another application server such as WebSphere Application Server Version 6, use the appropriate commands for that server.</p> <p>Usage</p> <pre>dsysync</pre> |
| dsysyncstatus | <p>This command returns the execution status of the DB2 Everyplace server that is installed in the embedded version of the WebSphere Application Server Version 6. The server can be started, running, or stopped. This command takes no arguments.</p> <p>If DB2 Everyplace is installed in another application server, use the appropriate commands for that server.</p> <p>Usage</p> <pre>dsysyncstatus</pre> |

Table 14. Commands for UNIX, Linux, and Windows (continued)

| Command | Description |
|-----------------|---|
| dsysyncstop | <p>This command stops the embedded version of the WebSphere Application Server Version 6 that executes the DB2 Everyplace servlets. This command takes no arguments.</p> <p>If DB2 Everyplace is installed in another application server, such as WebSphere Application Server Version 6, use the appropriate commands for that server.</p> <p>Usage
dsysyncstop</p> |
| dsytrace | <p>This command enables or disables tracing. By default, the DB2 Everyplace Sync Server logs error messages only in the trace files. However, for diagnostic purposes, you might want to enable tracing to include more detailed information.</p> <p>Usage
dsytrace <0 1> {-console}</p> <p>where</p> <p>0 Disables tracing except for errors.</p> <p>1 Enables tracing.</p> <p>-console
Disables tracing to the console window.</p> <p>Example: To enable tracing but not display it in the console window, issue the following command:
dsytrace 1 -console</p> |
| dsytraceprocess | <p>This command expands the timestamp format in the trace files to "yyymmdd hh:mm:ss.ms" format.</p> <p>Usage
dsytraceprocess -F<tracefileName> {-0 <>true false>}</p> <p>where</p> <p>tracefileName
the name of the trace file that you want to process</p> <p>-O the overwrite flag. If you specify true, the original trace file will be overwritten by the converted trace file with expanded timestamps. If you specify false, the resulting trace files will be named as originalTraceFile.new</p> <p>Important: There is no space between -F and <tracefileName>.</p> |

Table 14. Commands for UNIX, Linux, and Windows (continued)

| Command | Description |
|--------------------------|--|
| dsyunblocksubscription | <p>This command unblocks the subscription name that is provided in the command argument. The DB2 Everyplace Sync Server resumes accepting new changes from the mobile devices that are a part of the specified subscription.</p> <p>Usage</p> <pre>dsyunblocksubscription <subscription></pre> <p>where</p> <p><i>subscription</i> is the subscription to unblock on the server. You can specify multiple subscriptions.</p> <p>Example: To unblock the subscription with the name 'bob', issue the following command:</p> <pre>dsyunblocksubscription bob</pre> <p>Example: To unblock the subscriptions named 'bob' and 'sally', issue the following command:</p> <pre>dsyunblocksubscription bob sally</pre> |
| dsyuserexitstest | <p>This command tests user exit routines for a given DSY message ID.</p> <p>Usage</p> <pre>dsyuserexitstest <DSY message ID></pre> |
| dsyversion | <p>This command displays the version of DB2 Everyplace Sync Server.</p> <p>Usage</p> <pre>dsyversion</pre> |
| usev8dsydatatypemappings | <p>This command instructs the DB2 Everyplace Sync Server to use datatype mappings that are specific to Version 8.</p> <p>Usage:</p> <pre>usev8dsydatatypemappings</pre> |

Data type mappings between DB2 Everyplace and data sources

This topic shows the default mirror and client data types to which various backend data source data types are mapped.

Important:

1. Because of the inherent differences between non-DB2 data types and DB2 data types, the creation of certain subscriptions and the replication or synchronization of certain values might not be possible.
2. If a data type is not listed in the data type mapping tables, it is not supported.
3. You cannot include a table in a subscription if it contains a column of an unsupported data type.

Supported database default values

The following tables list the default values that a column of a table in a data source might have. A source table having columns with any of the default values listed can be synchronized by the DB2 Everyplace Sync Server. A table having columns with default values not listed in the table for the data source cannot be synchronized by the DB2 Everyplace Sync Server.

Note: If you are using the IBM Toolbox for Java driver to connect to DB2 on AS/400, your table cannot have any columns with non-null default values.

Table 15. Supported DB2 default values

| DB2 data type | DB2 default value |
|---------------------------|--------------------------|
| BIGINT | constant, NULL |
| CHAR(n) | constant, NULL |
| DATE | current date, NULL |
| DECIMAL(p,s) | constant, NULL |
| DOUBLE | constant, NULL |
| FLOAT | constant, NULL |
| GRAPHIC(n) | NULL |
| INTEGER | constant, NULL |
| LONG VARCHAR | constant, NULL |
| LONG VARCHAR FOR BIT DATA | NULL |
| LONG VARGRAPHIC | NULL |
| REAL | constant, NULL |
| SMALLINT | constant, NULL |
| TIME | current time, NULL |
| TIMESTAMP | current timestamp, NULL |
| VARCHAR(n) | constant, NULL |
| VARCHAR(n) FOR BIT DATA | NULL |
| VARGRAPHIC(n) | NULL |

Table 16. Supported Informix default values

| Informix data type | Informix default value |
|--|-------------------------------|
| CHAR | NULL |
| CHARACTER VARYING(m,r) | NULL |
| DATE | NULL |
| DATETIME HOUR TO SECOND | NULL |
| DATETIME HOUR TO FRACTION | NULL |
| DATETIME YEAR TO DAY | NULL |
| DATETIME YEAR TO SECOND | NULL |
| DATETIME YEAR TO FRACTION | NULL |
| DATETIME YEAR TO FRACTION(5) | NULL |
| DECIMAL(p,s) | NULL |
| DOUBLE PRECISION | NULL |
| FLOAT(n) | NULL |
| INT8 | NULL |
| INTEGER | NULL |
| INTERVAL largest_qualifier(p) TO smallest_qualifier(s) | NULL |
| LVARCHAR | NULL |
| MONEY(p,s) | NULL |

Table 16. Supported Informix default values (continued)

| | |
|--------------|------|
| NCHAR(n) | NULL |
| NUMERIC(p,s) | NULL |
| NVARCHAR(m) | NULL |
| REAL | NULL |
| SMALLFLOAT | NULL |
| SMALLINT | NULL |
| VARCHAR(m) | NULL |

Table 17. Supported Oracle default values

| Oracle data type | Oracle default value |
|------------------|----------------------|
| CHAR(n) | constant, NULL |
| DATE | SYSDATE, NULL |
| NUMBER(p,s) | constant, NULL |
| RAW(n) | NULL |
| TIMESTAMP | NULL |
| VARCHAR2(n) | constant, NULL |

Table 18. Supported Microsoft SQL Server default values

| Microsoft SQL Server data type | Microsoft SQL Server default value |
|--------------------------------|------------------------------------|
| BIGINT | constant, NULL |
| BIT | constant, NULL |
| CHAR | constant, NULL |
| DATETIME | NULL |
| DECIMAL | constant, NULL |
| FLOAT | constant, NULL |
| INTEGER | constant, NULL |
| MONEY | constant, NULL |
| NCHAR | constant, NULL |
| NUMERIC | constant, NULL |
| NVARCHAR | constant, NULL |
| REAL | constant, NULL |
| SMALLDATETIME | NULL |
| SMALLINT | constant, NULL |
| SMALLMONEY | constant, NULL |
| TINYINT | constant, NULL |
| VARCHAR | constant, NULL |

DB2™ family data type mappings

Table 19 on page 187 lists the data type mapping that is performed when the source data type is a DB2 Version 9.1 or DB2 Universal Database (UDB) Version 8.2 data type.

Table 19. Data type mapping from DB2 Universal Database source data types

| DB2 Version 9.1 and DB2 UDB Version 8.2 source data type | DB2 Version 9.1 mirror data type | DB2 Everyplace device data type | IBM Cloudscape Version 10 device data type |
|--|----------------------------------|---------------------------------|--|
| BIGINT | BIGINT | VARCHAR | BIGINT |
| BLOB(n [K M G]) | unsupported | unsupported | unsupported |
| CHAR(n) | CHARACTER | CHARACTER | CHARACTER |
| CHAR(n) FOR BIT DATA | unsupported | unsupported | unsupported |
| CLOB(n [K M G]) | unsupported | unsupported | unsupported |
| DATALINK | unsupported | unsupported | unsupported |
| DATE | DATE | DATE | DATE |
| DBCLOB(n [K M G]) | unsupported | unsupported | unsupported |
| DECIMAL(p,s) | DECIMAL | DECIMAL | DECIMAL |
| DOUBLE | FLOAT | VARCHAR | DOUBLE PRECISION |
| DOUBLE PRECISION | FLOAT | VARCHAR | DOUBLE PRECISION |
| FLOAT | FLOAT | VARCHAR | DOUBLE PRECISION |
| GRAPHIC(n) | GRAPHIC | CHARACTER | unsupported |
| INTEGER | INTEGER | INTEGER | INTEGER |
| LONG VARCHAR | LONG VARCHAR | VARCHAR | LONG VARCHAR |
| LONG VARCHAR FOR BIT DATA | LONG VARCHAR FOR BIT DATA | BLOB | LONG BIT VARYING |
| LONG VARGRAPHIC | LONG VARGRAPHIC | VARCHAR | unsupported |
| REAL | REAL | VARCHAR | REAL |
| SMALLINT | SMALLINT | SMALLINT | SMALLINT |
| TIME | TIME | TIME | TIME |
| TIMESTAMP | TIMESTAMP | TIMESTAMP | TIMESTAMP |
| VARCHAR(n) | VARCHAR | VARCHAR | VARCHAR |
| VARCHAR(n) FOR BIT DATA | VARCHAR() FOR BIT DATA | BLOB | BIT VARYING |
| VARGRAPHIC(n) | VARGRAPHIC | VARCHAR | unsupported |
| XML | unsupported | unsupported | unsupported |

Related reference

“Data type mappings between DB2 Everyplace and data sources” on page 184

This topic shows the default mirror and client data types to which various backend data source data types are mapped.

“Informix data type mappings”

“Oracle data type mappings” on page 189

Informix data type mappings

Table 20 on page 188 lists the data type mapping that is performed when the source data type is an Informix data type.

Table 20. Data type mapping from Informix source data types

| Informix source data type | DB2 Version 9.1 mirror data type | DB2 Everyplace device data type | IBM Cloudscape Version 10 device data type |
|--|----------------------------------|---------------------------------|--|
| BLOB | unsupported | unsupported | unsupported |
| BOOLEAN | unsupported | unsupported | unsupported |
| BYTE | unsupported | unsupported | unsupported |
| CHAR(n) | CHARACTER | CHARACTER | CHARACTER |
| CHARACTER VARYING(m,r) | VARCHAR | VARCHAR | VARCHAR |
| CLOB | unsupported | unsupported | unsupported |
| DATE | DATE | DATE | DATE |
| DATETIME HOUR TO SECOND | TIME | TIME | TIME |
| DATETIME HOUR TO FRACTION | TIMESTAMP | TIMESTAMP | TIMESTAMP |
| DATETIME YEAR TO DAY | DATE | DATE | DATE |
| DATETIME YEAR TO SECOND | TIMESTAMP | TIMESTAMP | TIMESTAMP |
| DATETIME YEAR TO FRACTION | TIMESTAMP | TIMESTAMP | TIMESTAMP |
| DATETIME YEAR TO FRACTION(5) | TIMESTAMP | TIMESTAMP | TIMESTAMP |
| DECIMAL(p,s) | DECIMAL | DECIMAL | DECIMAL |
| DOUBLE PRECISION | DECIMAL | DECIMAL | DECIMAL |
| FLOAT(n) | FLOAT | VARCHAR | FLOAT |
| INT8 | BIGINT | VARCHAR | BIGINT |
| INTEGER | INTEGER | INTEGER | INTEGER |
| INTERVAL largest_qualifier(p) TO smallest_qualifier(s) | CHARACTER | CHARACTER | CHARACTER |
| LVARCHAR | VARCHAR | VARCHAR | VARCHAR |
| MONEY(p,s) | DECIMAL | DECIMAL | DECIMAL |
| NCHAR(n) | CHARACTER | CHARACTER | CHARACTER |
| NUMERIC(p,s) | NUMERIC | DECIMAL | NUMERIC |
| NVARCHAR(m) | VARCHAR | VARCHAR | VARCHAR |
| REAL | REAL | VARCHAR | REAL |
| SERIAL(n) | unsupported | unsupported | unsupported |
| SERIAL8 | unsupported | unsupported | unsupported |
| SMALLFLOAT | REAL | VARCHAR | REAL |
| SMALLINT | SMALLINT | SMALLINT | SMALLINT |
| TEXT | unsupported | unsupported | unsupported |
| VARCHAR(m) | VARCHAR | VARCHAR | VARCHAR |

Related reference

“Data type mappings between DB2 Everyplace and data sources” on page 184

This topic shows the default mirror and client data types to which various backend data source data types are mapped.

“DB2™ family data type mappings” on page 186

“Oracle data type mappings”

Oracle data type mappings

Table 21 lists the data type mapping that is performed when the source data type is an Oracle data type.

Table 21. Data type mapping from Oracle source data types

| Oracle source data type | DB2 mirror data type | DB2 Everyplace device data type | IBM Cloudscape Version 10 device data type |
|-------------------------|------------------------|---------------------------------|--|
| BFILE | unsupported | unsupported | unsupported |
| BLOB | unsupported | unsupported | unsupported |
| CHAR(n) | CHARACTER | CHARACTER | CHARACTER |
| CLOB | unsupported | unsupported | unsupported |
| DATE | TIMESTAMP | TIMESTAMP | unsupported |
| FLOAT | unsupported | unsupported | unsupported |
| LONG | unsupported | unsupported | unsupported |
| LONG RAW | unsupported | unsupported | unsupported |
| NCHAR(n) | unsupported | unsupported | unsupported |
| NCLOB | unsupported | unsupported | unsupported |
| NUMBER(p,s) | DECIMAL | DECIMAL | DECIMAL |
| NVARCHAR2(n) | unsupported | unsupported | unsupported |
| RAW(n) | VARCHAR() BIT FOR DATA | BLOB | BIT VARYING |
| REAL | unsupported | unsupported | unsupported |
| ROWID | unsupported | unsupported | unsupported |
| TIMESTAMP | TIMESTAMP | TIMESTAMP | TIMESTAMP |
| UROWID | unsupported | unsupported | unsupported |
| VARCHAR2(n) | VARCHAR | VARCHAR | VARCHAR |

Related reference

“Data type mappings between DB2 Everyplace and data sources” on page 184

This topic shows the default mirror and client data types to which various backend data source data types are mapped.

“DB2™ family data type mappings” on page 186

“Informix data type mappings” on page 187

Microsoft SQL Server data type mappings

Table 22 on page 190 lists the data type mapping that is performed when the source data type is Microsoft SQL Server. In DB2 Everyplace version 8.1.4 and below, the mapping for Microsoft SQL Server BIT data type is inconsistent between JDBC and upload subscriptions. In JDBC subscriptions, the Microsoft SQL Server data type BIT is mapped to the DB2 Everyplace data type SMALLINT. In upload subscriptions, the Microsoft SQL Server data type BIT is mapped to the DB2 Everyplace data type VARCHAR(1). In DB2 Everyplace version 8.2, the BIT is mapped to SMALLINT in both cases. If you

desire the old, inconsistent behavior, run the following script and restart the DB2 Everyplace Sync Server:
 dsysetproperty "DatatypeMappings Generic Target:*" -7="12 VARCHAR"

Table 22. Data type mapping from Microsoft SQL Server

| Microsoft SQL Server source type | DB2 Version 9.1 mirror data type | DB2 Everyplace device data type | IBM Cloudscape Version 10 device data |
|----------------------------------|----------------------------------|---------------------------------|---------------------------------------|
| BIGINT | BIGINT | VARCHAR | BIGINT |
| BINARY | unsupported | unsupported | unsupported |
| BIT | SMALLINT | SMALLINT | SMALLINT |
| CHAR | CHARACTER | CHARACTER | CHARACTER |
| CURSOR | unsupported | unsupported | unsupported |
| DATETIME | TIMESTAMP | TIMESTAMP | TIMESTAMP |
| DECIMAL | DECIMAL | DECIMAL | DECIMAL |
| FLOAT | FLOAT | VARCHAR | FLOAT |
| IMAGE | unsupported | unsupported | unsupported |
| INT | INTEGER | INTEGER | INTEGER |
| MONEY | DECIMAL | DECIMAL | DECIMAL |
| NCHAR | GRAPHIC | CHARACTER | CHARACTER |
| NTEXT | unsupported | unsupported | unsupported |
| NUMERIC | DECIMAL | DECIMAL | DECIMAL |
| NVARCHAR | VARGRAPHIC | VARCHAR | VARCHAR |
| REAL | REAL | VARCHAR | REAL |
| SMALLDATETIME | TIMESTAMP | TIMESTAMP | TIMESTAMP |
| SMALLINT | SMALLINT | SMALLINT | SMALLINT |
| SMALLMONEY | DECIMAL | DECIMAL | DECIMAL |
| TEXT | unsupported | unsupported | unsupported |
| TIMESTAMP | unsupported | unsupported | unsupported |
| TINYINT | SMALLINT | SMALLINT | SMALLINT |
| UNIQUEIDENTIFIER | unsupported | unsupported | unsupported |
| VARBINARY | unsupported | unsupported | unsupported |
| VARCHAR | VARCHAR | VARCHAR | VARCHAR |

Related reference

- “Data type mappings between DB2 Everyplace and data sources” on page 184
 This topic shows the default mirror and client data types to which various backend data source data types are mapped.
- “DB2™ family data type mappings” on page 186
- “Informix data type mappings” on page 187
- “Oracle data type mappings” on page 189

Data type mapping restrictions

The following restrictions exist when data type mapping is performed:

- If DB2 mirror types GRAPHIC, VARGRAPHIC are used, the DB2 mirror database has to be created in DBCS.

- Mobile device applications must take care that the type of the data entered into a column of a device table is compatible with the types of the columns of the corresponding mirror and source tables and the length of the data does not exceed the length of the corresponding mirror and source columns. Source applications must ensure the same with data entered into a column of a source table.
- Informix columns of the type DECIMAL, NVARCHAR, and VARCHAR must not be defined using the following syntax: DECIMAL(p), NVARCHAR(m,r), and VARCHAR(m,r).
- The following data types are not supported as primary keys by DB2 Everyplace Sync Server:
 - **DB2 Version 9.1 Type:** LONG VARCHAR, LONG VARCHAR FOR BIT DATA, LONG VARGRAPHIC, and VARCHAR() FOR BIT DATA
 - **Informix Type:** DATETIME and INTERVAL
 - **Oracle Type:** RAW
 - **SQL Server Type:** MONEY, REAL, and SMALLMONEY

Because of the inherent differences between non-DB2 data types and DB2 data types, the creation of certain subscriptions and the replication or synchronization of certain values might not be possible. If a data type is not listed in the data type mapping tables, it is not supported.

Data source restrictions for DataPropagator subscriptions Restrictions

The following restrictions apply to both Windows and UNIX platforms:

- Before you can apply changes, you must first start the Capture program to capture the changes on the source database.
- Do not create a DataPropagator subscription on tables that have referential-integrity constraints or triggers. Otherwise, replication would fail and would be unrecoverable.
- A mirror database must only replicate with a single source database.
- For z/OS source systems, you must issue a bind Capture job to start the Capture program on the source system. For more information about how to bind the Capture program, see the DB2 Version 9.1 documentation.
- For iSeries:
 - You must start the Capture program on the source database before DB2 Everyplace Sync Server can be started. Use the STRDPRCAP command to start the Capture program.
 - Do not use the XML scripting tool to create control tables for the source database on iSeries because the tool does not support that function. In other words, do not set the **CreateDPropRControlTables** attribute of tag <AddReplMaster> to TRUE. You must create control tables manually by using the CRTDPRTBL command.
 - You must journal source tables manually. Use the STRJRNPf command to journal source tables.

Note: If you use the CREATE SCHEMA statement to create a library that contains the source tables, journaling will perform automatically.

For more information on the commands, see the iSeries Information Center and SQL Replication Guide and Reference.

- The mirror database must be located on the same server as the replication engine, which is a replication-enabled DB2 Everyplace Sync Server or a command-line process running the dsyreplicate script. The DataPropagator Capture program is not able to capture any remote databases.
- By default, DataPropagator will create the mirror tables in their own, non-system managed, table spaces if the subscription is created with Replication Center. The XML Scripting tool does not create the tables in their own non-system managed table spaces. If you want the DB2 Everyplace control tables associated with the mirror tables to be created in the same table spaces, you will need to either override the table space creation to specify that the table space is system-managed, or increase the table space container size to accommodate these control tables (a general recommendation is to increase the

container to 5 times the Data Propagator-generated size). If you do not specify a table space in the XML to generate the DB2 Everyplace Subscription, the default table space USERSPACE1 is used for the DB2 Everyplace control tables.

- If `Replicate = "FALSE"` is not specified as an attribute in the `AddDProprSubscription` tag, then the mirror database must be local to the machine on which the XML Scripting Tool is executing, and the source capture process must be running.
- By default, the `CommitCount` value for a DataPropagator table subscription is 0, which forces all replicating changes to be performed within a single transaction. If an error occurs during replication, all changes are rolled back, and when recovery is attempted, changes are not replayed. However, using this feature increases the amount of transaction space required, especially when an application generates many changes. Depending on the client environment and application requirements, you might need to set `CommitCount` to a positive value or increase the transaction log space to support a `CommitCount` value of 0. You can set the `CommitCount` value in the XML script that creates or alters the DataPropagator table subscription, or modify it using the DB2 Version 9.1 Replication Center.
- The DB2 Everyplace Sync Server does not synchronize times or timestamps with an hour of 24 correctly due to differences in how time is represented in Java and a data source. A time of "24:00:00", for example, is converted to "00:00:00" and causes the data saved in the mobile database to differ from the data saved in the source database. You should avoid using such an hour in your applications.
- If you are adding a table to a subscription, its source and target schema names, column names, and table names cannot be keywords, reserved words, or special registers in SQL or DB2 Version 9.1.
- A mirror database can replicate only with one source database. DB2 Everyplace does not allow a mirror database to be associated with multiple source databases.
- In rare cases, this can result in an inability to create a table due to non-uniqueness of the table name or column name. In other rare cases it can result in a conversion of a character to multiple characters, as in the German "sharp s" (ß), which will be converted to "SS".

DB2 Everyplace limits

Table 23 describes the limits for the DB2 Everyplace database and SQL support. If you adhere to the most restrictive case, your programs might be easier to port to other platforms. Some mobile devices might have further restrictions on these limits due to physical memory and system limitations. Refer to the documentation that came with your mobile device for more information about these limitations. Each constraint applies to all clients unless otherwise specified.

Table 23. DB2 Everyplace database and SQL limits

| Description | Limit |
|--|---------------------|
| Maximum combined length for INT, SMALLINT, CHAR, DECIMAL, DATE, TIME, and TIMESTAMP columns in a single record | 32767 bytes |
| Maximum length of a BLOB column | 2 Gigabytes -1 byte |
| Maximum length of a CHAR column | 32767 bytes |
| Maximum length of a SQL statement | 64 kilobytes |
| Maximum length of a VARCHAR column | 32767 bytes |
| Maximum length of a check constraint | 32767 bytes |
| Maximum length of a column name (Cloudscape 10.0 client) | 30 |
| Maximum length of a column name (Other clients) | 128 |
| Maximum length of a default value | 32767 bytes |
| Maximum length of a row in a table | 64 kilobytes |
| Maximum length of a table name | 128 |
| Maximum length of an identifier | 128 |

Table 23. DB2 Everyplace database and SQL limits (continued)

| Description | Limit |
|--|-----------------------|
| Maximum length of an index name (Cloudscape 10.0) | 18 |
| Maximum length of an index name (Other clients) | 128 |
| Maximum length of each column in a single index | 1024 bytes |
| Maximum number of columns in a foreign key | 8 |
| Maximum number of columns in an index | 8 |
| Maximum number of columns in a primary key | 8 |
| Maximum number of columns in a table | 256 |
| Maximum number of indices in a table | 15 |
| Maximum number of LOB locators | 256 |
| Maximum number of rows in a table | Limited by table size |
| Maximum number of statement handles per connection | 256 |
| Maximum number of tables in a data store | 65535 |
| Maximum size of a decimal | 31 digits |
| Maximum size of a literal | 32672 bytes |
| Maximum size of a table (on a 32 bit system) | 2 Gigabytes |
| Maximum year for a date value | 9999 |
| Minimum year for a date value | 0001 |

DB2 Everyplace has additional limits about the size of fields that it synchronizes with the DB2 Everyplace Sync Server. If the synchronization size limit is not shown in the table below, it is the same size as the limit that is shown in Table 23 on page 192. Each constraint applies to all clients unless otherwise specified.

Table 24. DB2 Everyplace synchronization limits

| Description | Limit |
|---|-------|
| Maximum length of a column name | 30 |
| Maximum length of a table name (Cloudscape 10.0 and 10.1 clients) | 126 |

Special limitations for the length of table names

Restriction: The actual length of table names might be less than 128 characters due to expansion after the name is converted to UTF-8.

Starting in DB2 Everyplace version 9.1, the maximum identifier length for table names, column names, and user names has been increased from 18 to 128 for newly created databases. If an older database exists, the maximum identifier length remains at 18.

Restriction: Previous versions of DB2 Everyplace do not support table names that are as long as DB2 Everyplace version 9.1. In order to create tables that have long names, you must create the database in DB2 Everyplace version 9.1. Otherwise, DB2 Everyplace limits the length of table names to the maximum length of the version with which the table was created.

The following platforms cannot support the 128 character limit:

Palm OS

Table names on Palm OS mobile devices cannot be longer than 26 characters. This is because Palm OS limits filenames to 31 characters, and DB2 Everyplace appends the 5 characters "DSY_i" for the index of each table.

QNX Neutrino version 6.2 and earlier

Table names cannot be longer than 43 characters. This is because QNX Neutrino limits filenames to 48 characters, and DB2 Everyplace appends the 5 characters "DSY_i" for the index of each table.

QNX Neutrino version 6.2.1 and newer

To avoid the 43 character table size limit:

1. Log in as the root user.
2. Change to the root directory.
3. Create an empty, read-only file named .longfilenames.
4. Reboot.

Overview of the DB2 Everyplace mobile database tables

A DB2 Everyplace mobile database comprises several system catalog tables and a number of user-defined tables.

Each table is stored in two files: one for the data itself, and one for indexes. All indexes are kept in the same index file. Unlike DB2 Version 9.1, DB2 Everyplace mobile databases do not have names and cannot be cataloged or uncataloged. Therefore, the database name is ignored.

A DB2 Everyplace mobile database is a set of files that can be copied or moved to another location. A DB2 Everyplace mobile database must contain the following system catalog tables:

- DB2eSYSTABLES
- DB2eSYSCOLUMNS
- DB2eSYSRELS
- DB2eSYSUSERS (this table is created if you use local data encryption)

System catalog tables contain metadata about user-defined tables. For example, if you remove files for a user-defined table without deleting a corresponding entry in the catalog tables, you will cause an inconsistency.

To access catalog tables in a query, you must use delimited identifiers. For example, the following query returns 1 if the table T exists:

```
SELECT 1 FROM "DB2eSYSTABLES" WHERE TNAME = 'T'
```

Related reference

"DB2 Everyplace System Catalog base tables"

The database manager creates and maintains a set of system catalog base tables. This appendix contains a description of each system catalog base table, including column names and data types.

DB2 Everyplace System Catalog base tables

The database manager creates and maintains a set of system catalog base tables. This appendix contains a description of each system catalog base table, including column names and data types.

All of the system catalog base tables are created by the database manager. The system catalog base tables cannot be explicitly created or dropped. The system catalog base tables are updated during normal operation in response to SQL data definition statements, environment routines, and certain utilities. Data in the system catalog base tables is available through normal SQL query facilities. The system catalog

base tables cannot be modified using normal SQL data manipulation commands. In order to access the system catalog tables, you need to use a delimited identifier.

Table 25. System catalog base tables

| Description | Catalog base table |
|-------------------------|----------------------------|
| tables | "DB2eSYSTABLES" |
| columns | "DB2eSYSCOLUMNS" |
| referential constraints | "DB2eSYSRELS" on page 196 |
| users | "DB2eSYSUSERS" on page 196 |

DB2eSYSTABLES

This system catalog base table contains one row for each table that is created. All of the catalog tables have entries in the DB2eSYSTABLES catalog.

Table 26. DB2eSYSTABLES system catalog base table

| Column name | Data type | Nullable | Description |
|-------------|---------------|----------|---|
| TNAME | VARCHAR (129) | | Table name |
| NUMCOLS | INTEGER (4) | | Number of columns |
| FLAGS | INTEGER (4) | | (Internal use only) |
| NUMKEY | INTEGER (4) | | Number of columns in the primary key |
| CHK | BLOB (32767) | Yes | Check constraint (internal use only) |
| IDXINFO | BLOB (4096) | Yes | Index (internal use only) |
| NUMREFS | INTEGER (4) | Yes | Primary and foreign key (internal use only) |
| F_ID | INTEGER (4) | Yes | (Internal use only) |
| PD | BLOB (4096) | Yes | (Internal use only) |

DB2eSYSCOLUMNS

This system catalog base table contains one row for each column that is defined for a table.

Table 27. DB2eSYSCOLUMNS system catalog base table

| Column name | Data type | Nullable | Description |
|-------------|-----------------|----------|---|
| CNAME | VARCHAR (129) | | Column name |
| TNAME | VARCHAR (129) | | Table name |
| TYPE | INTEGER (4) | | Data type |
| ATTR | INTEGER (4) | | (Internal use only) |
| LENGTH | INTEGER (4) | | Length of the column |
| POS | INTEGER (4) | | Column number |
| FLAGS | INTEGER (4) | | (Internal use only) |
| KEYSEQ | INTEGER (4) | | Ordinal position of the column in the primary key |
| SCALE | INTEGER (4) | | Scale for decimal column |
| DEF | VARCHAR (32767) | Yes | Default value (internal use) |

DB2eSYSRELS

This system catalog base table contains a row for each referential constraint.

Table 28. DB2eSYSRELS system catalog base table

| Column name | Data type | Nullable | Description |
|------------------|---------------|----------|---|
| RMD_ID | INTEGER (4) | | Primary and foreign key (internal use only) |
| PKTABLE_NAME | VARCHAR (129) | | Parent table name |
| PKCOLUMN_NAME | VARCHAR (129) | | Parent table primary key column |
| FKTABLE_NAME | VARCHAR (129) | | Child table name |
| FKCOLUMN_NAME | VARCHAR (129) | | Child table foreign key column name |
| ORDINAL_POSITION | INTEGER (4) | | Position of the column in the foreign key reference |

DB2eSYSUSERS

The DB2eSYSUSERS table is created automatically when the first encrypted table is created or when the first GRANT statement is executed. This table is tightly bound to the database and encrypted data; it cannot be moved to another DB2 Everyplace database that contains different encrypted data.

This system catalog base table contains one row for each registered user name that is defined for a database.

Table 29. DB2eSYSUSERS system catalog base table

| Column name | Data type | Nullable | Description |
|--------------|----------------|----------|--|
| USERNAME | VARCHAR (129) | | Part of primary key and is case sensitive. The name of the user associated with this row. |
| DATABASENAME | VARCHAR (129) | | For future use. Empty string is stored. Part of primary key. |
| TABlename | VARCHAR (129) | | For future use. Empty string is stored. Part of primary key. |
| ENCMETHOD | VARCHAR (129) | | For future use. Empty string is stored. Part of primary key. |
| PRIVILEGES | VARCHAR (129) | Yes | Defines user privileges. Currently, only the value 'E', indicating encryption, is allowed. |
| ENCKEYDATA | BLOB (280) | Yes | Used to regenerate encryption key. |
| ATTIME | TIMESTAMP (26) | Yes | Time when the user was added or the record was most recently modified, whichever is most recent. |
| VALIDATE | BLOB (280) | Yes | Verifies that the record is authentic and the user was added by an authenticated user. |
| GRANTOR | VARCHAR (129) | Yes | The user name that registered the user name in column 1. |
| INTERNALDATA | BLOB (255) | Yes | (Internal future use) |

DB2 Everyplace error messages

This topic explains error messages that are returned by DB2 Everyplace and its subsystems.

Error messages

This topic lists the error messages and their meanings, and it provides suggested actions to correct the problem identified in the message. Use this information with “Handling DB2 Everyplace synchronization problems” on page 157 to identify and correct problems.

DSYA002E: User *user name* attempt to register device *device id* failed as the device is already registered to user *registered user name*. Action canceled.

Explanation: The DB2 Everyplace control database could not assign the device to the specified user as the device is currently assigned to the specified registered user.

User Response: If the device should be registered to the specified user, use the DB2 Everyplace Mobile Devices Administration Center to delete the device associated with the registered user and try again.

DSYC2000C: Requested data changes rejected.

Explanation: Some of the data changes that the DB2 Everyplace Sync Client requested have been rejected due to data conflicts or no permission.

User Response: Check with the DB2 Everyplace Sync Server administrator to ensure that the user has permission to perform the desired operation. Because the requested changes have been rejected, the user has to re-apply the changes, or have the DB2 Everyplace Sync Server administrator reset the user and try again. If the problem persists, contact IBM software support.

DSYC300E: Failed to open adapter: *adapter name*.

Explanation: The DB2 Everyplace Sync Client could not find the adapter library for synchronizing a subscription.

User Response: Check that the library ‘*adapter name*’ is present on the device. If the library name is unknown, contact the DB2 Everyplace Sync Server administrator, and then have the administrator reset the user and try again. If the problem persists, contact IBM software support.

DSYC301E: Failed to load adapter: *adapter name*.

Explanation: The DB2 Everyplace Sync Client could not load the adapter library for synchronizing a subscription.

User Response: Check that the library for ‘*adapter name*’ is present on the device. If the library does exist in the device, it might be because the operating system has reached its limits of the opened shared libraries. Close unused applications, or restart the DB2 Everyplace Sync Client application, and/or do a soft reset on the device, and then try again. If the library name is unknown, contact the DB2 Everyplace Sync Server administrator, and then have the administrator reset the user and try again. If the problem persists, contact IBM software support.

DSYC302E: Failed to close adapter: *adapter name*.

Explanation: The DB2 Everyplace Sync Client could not close the adapter library after synchronization.

User Response: The library for ‘*adapter name*’ have been used and locked by other DB2 Everyplace Sync Client applications. Check if there are any other pending DB2 Everyplace Sync Client applications, and/or do a soft reset on the device, and then try again. If the problem persists, contact IBM software support.

DSYC303E: Configuration synchronization failed, synchronization aborted.

Explanation: The DB2 Everyplace Sync Client could not properly synchronize the configuration information.

User Response: Contact the DB2 Everyplace Sync Server Administrator to check the health of the DB2 Everyplace Sync Server. Restart the DB2 Everyplace Sync Server, and/or do a soft reset on the user, and then try again. If problem persists, contact IBM software support.

DSYC304E: Authentication failed - synchronization aborted.

Explanation: The provided username/password does not pass the authentication on the DB2 Everyplace Sync Server.

User Response: Make sure that the password is entered correctly, and try again. If problem persists, contact the DB2 Everyplace Sync Server administrator.

DSYC306E: Unrecognizable client message format

Explanation: The DB2 Everyplace Sync Server does not recognize the format of the client message.

User Response: Make sure that encryption is supported on the DB2 Everyplace Sync Client platform, and/or do a soft reset on the device, and then try again. If problem persists, contact IBM software support.

DSYC307E: Client encryption/decryption failed.

Explanation: The DB2 Everyplace Sync Client could not successfully encrypt the outgoing message or decrypt the received message.

User Response: Make sure that the client's operating system supports encryption and the chosen encryption level, do a soft reset on the device and then try again. If problem persists, contact IBM software support.

DSYC308E: Encryption not available.

Explanation: Encryption is not supported for Palm OS 3.2 and earlier. The encryption library not be installed properly or the encryption path is not set correctly.

User Response: Make sure that the client's operating system supports encryption and the chosen encryption level, the encryption library is properly installed, and the encryption path is set correctly. Please refer to the Installation of User's Guide for the instruction of installation. If problem persists, contact the DB2 Everyplace Sync Server administrator.

DSYC309E: Failed to open the encryption library.

Explanation: The encryption library not be installed properly or the encryption path is not set correctly.

User Response: Make sure that the client's operating system supports encryption and the chosen encryption level, the encryption library is properly installed, and the encryption path is set correctly. Please refer to the Installation of User's Guide for the instruction of installation. If problem persists, contact the DB2 Everyplace Sync Server administrator.

DSYC310E: Incompatible DB2 Everyplace Sync Client version.

Explanation: The version for the DB2 Everyplace Sync Client is not compatible with the DB2 Everyplace Sync Server.

User Response: Check with the DB2 Everyplace Sync Server administrator to make sure a compatible DB2 Everyplace Sync Client version is installed on the device. If the problem persists, contact IBM software support.

DSYC311E: Subscription not found.

Explanation: The DB2 Everyplace Sync Server cannot find the subscription that the client is requesting. It be because the configuration has changed because the client starts synchronization.

User Response: Try to synchronize again, and the configuration will be updated. If the problem still occurs, contact the DB2 Everyplace Sync Server administrator regarding the problematic subscription, and/or reset the user, then synchronize again. If the problem persists, contact IBM software support.

DSYC312E: Subscription blocked by the Server.

Explanation: The subscription has been blocked by the DB2 Everyplace Sync Server. It be because the DB2 Everyplace Sync Server administrator is performing maintenance on the control database or modifying the subscription, or the DB2 Everyplace Sync Server is replicating the subscription, or the last replication failed, but has not yet been recovered.

User Response: Please wait for a while and try to synchronize again. If the problem persists, contact the DB2 Everyplace Sync Server administrator to check the replication status of the subscription.

DSYC313E: Attempt to synchronize from a backed-up client

Explanation: The server has detected and rejected the attempt to synchronize from a backed-up client.

User Response: Please make sure the current client image (including both the configuration and synchronized data) is good and wanted. If yes, select the "Allow synchronization from backup" check box; otherwise, have the DB2 Everyplace Sync Server administrator reset the user. Then, try to synchronize again. If the problem persists, contact the DB2 Everyplace Sync Server administrator.

DSYC314E: User not assigned to any group with synchronization privilege

Explanation: The user has not been added to any group with synchronization privilege.

User Response: Check that the username is entered correctly and contact the DB2 Everyplace Sync Server administrator to verify that the user is in a group with synchronization enabled. Try to synchronize again. If the problem persists, contact IBM software support.

DSYC315E: Failed to register the device.

Explanation: The DB2 Everyplace Sync Server cannot register the device with the user. There be already a device registered with the user, while the user is configured to allow to synchronize from one device.

User Response: Check that the username is entered correctly and contact the DB2 Everyplace Sync Server administrator to check the user association of the device, reset the user, and then try to synchronize again. If the problem persists, contact IBM software support.

DSYC316E: Subscription definition altered.

Explanation: The definition of a subscription has changed because the client starts synchronization.

User Response: Try to synchronize again, and the subscription definition will be updated. If the problem still occurs, contact the DB2 Everyplace Sync Server administrator to reset the user, and synchronize again. If the problem persists, contact IBM software support.

DSYC317E: Attempt to create client image without a valid device ID

Explanation: The DB2 Everyplace Sync Client engine cannot create a client image without a valid deviceID.

User Response: Creating a client image should be performed by the administrator. You can only create a client image from a client which has successfully synchronized before and has a valid device ID. Please first synchronize with the "CreateImage" option off, then try to create the client image again. If the problem persists, contact IBM software support.

DSYC400E: Failed to allocate adapter resources.

Explanation: The DB2 Everyplace Sync Client could not allocate adapter-specific resources (for example, database connections, or statement handles for DB2 Everyplace) for synchronizing the subscribed data.

User Response: Check that the required adapter-specific resources are not occupied and try again. If the problem persists, consult the DB2 Everyplace Application Development Guide for the limitation on those resources or contact IBM software support.

DSYC401E: Failed to connect to the target data.

Explanation: The DB2 Everyplace Sync Client detected problems when connecting to or accessing the target data.

User Response: Check the access rights and existence of the resource or make sure the resource is not being used by another application. If synchronizing encrypted tables, make sure the provided DB2 Everyplace username and password have been granted the encryption privilege. Have the DB2 Everyplace Sync Server administrator reset the user and try again. If the problem persists, contact IBM software support.

DSYC402E: Failed to disconnect from the target data.

Explanation: The DB2 Everyplace Sync Client detected problems disconnecting or releasing the target data.

User Response: Check the access rights and existence of the resource or make sure the resource is not being used by another application. Have the DB2 Everyplace Sync Server administrator reset the user and try again. If the problem persists, contact IBM software support.

DSYC403E: No data subscribed in subscription

Explanation: The DB2 Everyplace Sync Client received unexpected empty or missing subscription information (for example, no subscribed table information) from the server.

User Response: There be missing information in the subscription. Contact the DB2 Everyplace Sync Server administrator.

DSYC412E: Unexpected message format.

Explanation: The DB2 Everyplace Sync Client does not recognize some message information from the server during synchronization of a subscription.

User Response: Have the DB2 Everyplace Sync Server administrator reset the user and try again. If the problem persists, contact IBM software support.

DSYC413E: Requested target data not found.

Explanation: The DB2 Everyplace Sync Client cannot find the target data (for example, table not found) for synchronization.

User Response: The target data does not exist, which might have been removed by other applications. To restore the data of the subscription, reset the containing subscription set, and synchronize again. If the problem persists, contact IBM software support.

DSYC414E: Unexpected end of data.

Explanation: The DB2 Everyplace Sync Client has encountered unexpected missing or end-of-data during synchronization of a subscription.

User Response: Contact IBM software support.

DSYC415E: Data too long for corresponding field.

Explanation: The DB2 Everyplace Sync Client received data that was too large (for example, over-sized file). Data have been truncated.

User Response: The data from the DB2 Everyplace Sync Server be invalid. Check with the DB2 Everyplace Sync Server administrator.

DSYC417E: Server reported subscription not enabled for synchronization.

Explanation: The user is not enabled to synchronize the subscription.

User Response: Make sure that the username is entered correctly, and contact the DB2 Everyplace Sync Server administrator to make sure the user is enabled to synchronize the subscription, and try to synchronize again. If the problem persists, contact IBM software support.

DSYC418E: Server reported exceptions.

Explanation: The DB2 Everyplace Sync Server has encountered problems (or exceptions) synchronizing the subscription.

User Response: Contact the DB2 Everyplace Sync Server administrator to check the health of the DB2 Everyplace Sync Server or the status of the source data.

DSYC420E: Attempt to upload changes to read-only data.

Explanation: The client attempts to upload changes to the target data which is configured as read-only in the subscription definition.

User Response: Contact the DB2 Everyplace Sync Server administrator to check the permission to the target data in the subscription definition. If the subscribed data is configured read-only, the user shouldn't change the target data. To refresh the target data, reset the subscription set, and synchronize again. If the problem persists, contact IBM software support.

DSYC421E: Attempt to upload impermissible operations.

Explanation: The client attempts to upload changes to the target data, which are not permissible in the subscription definition.

User Response: Contact the DB2 Everyplace Sync Server administrator to check the permission to the target data in the subscription definition. The user shouldn't perform impermissible operations to the target data. To refresh the target data, reset the subscription set, and synchronize again. If the problem persists, contact IBM software support.

DSYC423E: Not authorized to access the target data.

Explanation: The user is not authorized to access the requested data.

User Response: Check that the username and password (to access the target data) are entered correctly, and that the user is authorized to access the requested data. Have the DB2 Everyplace Sync Server administrator reset the user and try to synchronize again. If the problem persists, contact the DB2 Everyplace Sync Server administrator.

DSYC424E: Requested data not available.

Explanation: The requested data is not available. The target data is being used by another application.

User Response: Check that the DB2 Everyplace Sync Client has permission to modify the target data, and that it is not being used by another application. Have the DB2 Everyplace Sync Server administrator reset the user and try again. If the problem persists, contact IBM software support.

DSYC425E: Requested data type is not supported.

Explanation: The subscription adapter does not support the requested data type. For example, you can only synchronize .prc and .pdb files to a Palm device and there are some column types which are not supported by DB2 Everyplace.

User Response: Have the DB2 Everyplace Sync Server administrator verify that all the subscribed data are supported (for example, file type and column types), and try to synchronize again. If the problem persists, contact IBM software support.

DSYC426E: Invalid subscription target path.

Explanation: The user-provided subscription target path is either not a valid path name, or the path is not relative to the device target path.

User Response: Make sure the specified subscription target path is a valid path name and relative to the device target directory.

DSYC600E: Failed to open connection.

Explanation: The DB2 Everyplace Sync Client has a problem opening a connection with the DB2 Everyplace Sync Server because either you have a wrong server URL (Uniform Resource Locator), or the server is not up.

User Response: Check to ensure: 1)Server URL is correctly entered 2)The DB2 Everyplace Sync Server is currently operational. Try to synchronize again. If the problem persists, contact IBM software support.

DSYC601E: Failed to establish connection.

Explanation: The DB2 Everyplace Sync Client has a problem opening connection with the DB2 Everyplace Sync Server. This could be due to: 1)Wrong server URL (Uniform Resource Locator) 2)Server is not up 3)Network is busy 4)Network connection is not operational.

User Response: Check to ensure: 1)Server URL is correctly entered 2)The DB2 Everyplace Sync Client has access to the DB2 Everyplace Sync Server through either a serial or USB cable, a modem, or a network connection. 3)The DB2 Everyplace Sync Server is currently operational. 4)If using Windows RAS, or some PPP connection software (for example, Mocha PPP for the serial port, Softick PPP for the USB port), make sure the PPP connection between the mobile device and the desktop computer is established. 5)Try to synchronize again. If the problem persists, contact IBM software support.

DSYC602E: Failed to send request.

Explanation: The DB2 Everyplace Sync Client has successfully connected to the DB2 Everyplace Sync Server, but has a problem sending requests to the server. The causes might be: 1)Server is not up 2)Network connection is lost.

User Response: Check to ensure: 1)The DB2 Everyplace Sync Client has access to the DB2 Everyplace Sync Server through either a serial or USB cable, a modem, or a network connection. 2)The DB2 Everyplace Sync Server is currently operational. 3)If using Windows RAS, or some PPP connection software (for example, Mocha PPP for the serial port, Softick PPP for the USB port), make sure the PPP connection between the mobile device and the desktop computer is established. 4)Try to synchronize again. If the problem persists, contact IBM software support.

DSYC603E: Failed to receive reply.

Explanation: The DB2 Everyplace Sync Client has successfully connected to the DB2 Everyplace Sync Server, but has a problem receiving messages from the server, or the received message is corrupted or in an unexpected format. The causes could be: 1)Server is not up 2)Network connection is lost.

User Response: Check to ensure: 1)The DB2 Everyplace Sync Server is currently operational. 2)If using Windows RAS, or some PPP connection software (for example, Mocha PPP for the serial port, Softick PPP for the USB port), make sure the PPP connection between the mobile device and the desktop computer is established. 3)Try to synchronize again. If the problem persists, contact IBM software support.

DSYC604E: Timeout while receiving reply.

Explanation: The DB2 Everyplace Sync Client has successfully connected sent a request to the DB2 Everyplace Sync Server, but timed out while receiving the server message. Either the server needs more time preparing the acquired information, the network is busy, the server is not up, or the network connection is lost.

User Response: Check to ensure: 1)The DB2 Everyplace Sync Server is currently operational. 2)If using Windows RAS, or some PPP connection software (for example, Mocha PPP for the serial port, Softick PPP for the USB port), make sure the PPP connection between the mobile device and

the desktop computer is established. 3)Length the network timeout. 4)Try to synchronize again. If the problem persists, contact IBM software support.

DSYC605E: Failed to receive acknowledgment.

Explanation: The DB2 Everyplace Sync Client has successfully sent a request and received a reply from the DB2 Everyplace Sync Server, but does not have received an acknowledgement from the server. Either the server needs more time preparing the inquired information, the network is busy, the server is not up, or the network connection is lost.

User Response: Check to ensure: 1)The DB2 Everyplace Sync Server is currently operational. 2)If using Windows RAS, or some PPP connection software (for example, Mocha PPP for the serial port, Softick PPP for the USB port), make sure the PPP connection between the mobile device and the desktop computer is established. 3)Try to synchronize again. If the problem persists, contact IBM software support.

DSYC606E: Failed to open the Network library.

Explanation: The DB2 Everyplace Sync Client has a problem opening the Network library.

User Response: Check to ensure: 1) The Network library is present. 2) If using the PalmOS emulator, make sure the check box under Settings->Properties is set (checked). 3) Try a soft reset on the device, and synchronize again. If the problem persists, contact IBM software support.

DSYC608E: Failed to close the Network library.

Explanation: The DB2 Everyplace Sync Client has a problem closing the Network library.

User Response: The Network library be in a corrupted state. Do a soft reset on the device, and then synchronize again. If the problem persists, contact IBM software support.

DSYC609E: Failed to resolve hostname.

Explanation: The DB2 Everyplace Sync Client cannot resolve the IP for the provided hostname in the server URL (Uniform Resource Locator).

User Response: Make sure the server hostname is correctly specified.

DSYC610E: Out of memory.

Explanation: The DB2 Everyplace Sync Client could not allocate sufficient memory to complete synchronization.

User Response: The device could be low on available memory or the dynamic heap memory. Close out or delete some unused applications, and try again. If the problem persists, consult the DB2 Everyplace documentation for system requirements or contact IBM software support.

DSYC611E: Forbidden to synchronize to the server.

Explanation: The user is not allowed to synchronize to the server.

User Response: Contact the DB2 Everyplace Sync Server administrator to check if the user is authorized to synchronize to the DB2 Everyplace Sync Server.

DSYC612E: Server not found.

Explanation: The server cannot be found because either a wrong server URL (Uniform Resource Locator) is being used, the server is not up, or the DB2 Everyplace Sync Server is not installed properly.

User Response: Check to ensure: 1)Server URL (Uniform Resource Locator) is correctly entered; 2)The DB2 Everyplace Sync Server is currently operational; 3)The DB2 Everyplace Sync Server is installed properly. Try to synchronize again. If the problem persists, contact the DB2 Everyplace Sync Server administrator.

DSYC613E: Internal DB2 Everyplace Sync Server error.

Explanation: An internal error has occurred to the DB2 Everyplace Sync Server.

User Response: Contact the DB2 Everyplace Sync Server administrator to check the health of the DB2 Everyplace Sync Server. Restart the DB2 Everyplace Sync Server and try to synchronize again. If the problem persists, contact IBM software support.

DSYC614E: Server not responding.

Explanation: No servers available to service the client requests. All servers are either busy or blocked.

User Response: Contact the DB2 Everyplace Sync Server administrator to check the health and the availability of the DB2 Everyplace Sync Server. Try to synchronize again when the server is less busy. If the problem persists, contact IBM software support.

DSYC615E: Transport protocol not supported

Explanation: The transport protocol specified in the server URL (Uniform Resource Locator) is not supported.

User Response: Make sure the protocol specified in the server URL is supported. Currently, the only supported protocol is HTTP (HyperText Transfer Protocol). Use HTTPS if both DB2 Everyplace Sync Server and DB2 Everyplace Sync Client have been configured for SSL (Secure Socket Layer).

DSYC616E: Server busy

Explanation: No servers available to service the client requests. All servers are busy.

User Response: Contact the DB2 Everyplace Sync Server administrator to check the availability of the DB2 Everyplace Sync Server. Try to synchronize again when the server is less busy. If the problem persists, contact IBM software support.

DSYC698E: Internal error or inconsistent state.

Explanation: The subscription adapter encountered errors or an inconsistent state the target data manager during synchronization.

User Response: Check the DB2 Everyplace Sync Client configuration log, and identify the subscription type (config, table, or file). If it is a table subscription, consult the DB2 Everyplace Application Development Guide with the provided SQL state in the error message. Contact the DB2 Everyplace Sync Server administrator regarding the error state. If the problem persists, contact IBM software support.

DSYC699E: Unknown network error.

Explanation: The DB2 Everyplace Sync Client has encountered an unknown error while communicating with the DB2 Everyplace Sync Server.

User Response: Contact IBM technical support.

DSYD006E: The MDSS session monitor encountered an exception from the SQL statement: *SQL statement*.

Explanation: The DB2 Everyplace Sync Server records information about the state of each synchronization session in session monitoring tables in the administration control database, DSYCTLDB. The DB2 Everyplace Sync Server issues an SQL statement to add an entry for each new session so that session state information is persistent. The SQL statement failed because the session monitor tables could not be accessed.

User Response: Ensure that the DSYCTLDB database is functional and the storage is not exhausted. If no problems are found, contact IBM Software Support.

DSYD007E: The MDSS connection pool encountered the exception: *exception details*.

Explanation: The DB2 Everyplace Sync Server creates a pool of database connections for each database accessed. In this case, the DB2 Everyplace Sync Server attempted to use a connection from this pool for the named database failed. A possible explanation for this error is because the named database is not functional.

User Response: Ensure that the named database is functional. If no problems are found, contact IBM Software Support.

DSYD010E: The DB2 adapter failed to generate a DataPropagator password file for the Apply qualifier *apply qualifier*.

Explanation: Each time you start the DB2 Everyplace Sync Server, a password file is generated for each Apply qualifier. The DB2 Data Propagator uses the user ID and password in this file to access the source database. Synchronization fails for all subscriptions whose Apply qualifier does not have a corresponding password file. File creation fails because of inadequate storage in the directory where the DB2 Everyplace Sync Server is running.

User Response: First, ensure that adequate storage is available in the directory where the DB2 Everyplace Sync Server is running. Then, stop and restart the DB2 Everyplace Sync Server to attempt to generate the password file. If problems persist, contact IBM Software Support.

DSYD011E: The DB2 adapter was unable to create the tables required for synchronizing table *schema name.table name* in database *database name*.

Explanation: The DB2 Everyplace Sync Server creates staging tables for each mirror table involved in synchronization. These tables are created when you create a subscription against the database that corresponds to the mirror referenced in the message. Inadequate storage or a non-functional database causes the creation of these tables to fail.

User Response: Ensure that the database referenced in the message is functional and that adequate storage is available. Open the Mobile Devices Administration Center and remove and recreate the subscriptions for the referenced database to attempt to create the staging tables again. If you continue to receive this message, contact IBM Software Support.

DSYD012E: The DB2 adapter was unable to drop the tables required for synchronizing table *schema name.table name* in database *database name*.

Explanation: To manage synchronization, the DB2 Everyplace Sync Server creates multiple tables in association with a given mirror database. When you delete subscription sets associated with the mirror in the named database, the DB2 Everyplace Sync Server drops the tables associated with the deleted subscription set. This drop operation might fail if the database is not functional.

User Response: Ensure that the database named in the message is functional.

DSYD014E: The DB2 adapter was unable to access the synchronization mirror table *schema name.table name* in database *database name*.

Explanation: An error occurred while attempting to access the mirror table.

User Response: Ensure that you are connected to the mirror database named in the message and that the database has adequate storage capacity. If problems persist, contact IBM Software Support.

DSYD015E: The DB2 adapter was unable to access the synchronization tables peripheral to mirror table *schema name.table name* in database *database name*.

Explanation: To manage synchronization, the DB2 Everyplace Sync Server creates multiple tables in association with a given mirror table. An error occurred when the DB2 Everyplace Sync Server attempted to access these tables for the mirror table referenced in the message.

User Response: Ensure that the database named in the message is functional. If problems persist, contact IBM Software Support.

DSYD018E: The DB2 adapter was unable to access the ASN.IBMSNAP_APPLYTRAIL table in database *database name*.

Explanation: An error occurred while attempting to access the DataPropagator apply trail table.

User Response: Ensure that the database named in the message is functional, and that the DB2 Data Propagator subscription was properly set up. In addition, see the DB2 Replication Guide and Reference for more information about why the Apply Trail table could not be accessed. If problems persist, contact IBM Software Support.

DSYD019E: The DB2 adapter was unable to access the ASN.IBMSNAP_UOW table in database *database name*.

Explanation: An error occurred while attempting to access the DataPropagator unit-of-work table.

User Response: Ensure that the database named in the message is functional, and that the Data Propagator subscription was properly set up. Additionally, see the DB2 Replication Guide and Reference for more information on why the unit-of-work table could not be accessed. If problems persist, contact IBM Software Support.

DSYD022E: The DB2 adapter was unable to access the change data table *schema.table name* in database *database name*.

Explanation: An error occurred while attempting to access the Data Propagator change data table.

User Response: Ensure that the database named in the message is functional, and that the subscription is properly set up in the Mobile Devices Administration Center. If problems persist, contact IBM Software Support.

DSYD027E: The DataPropagator Apply process failed for database *database name*, Apply qualifier *apply qualifier*.

Explanation: Synchronization of this database is not possible until the DataPropagator Apply error is addressed.

User Response: See the DB2 Replication Guide and Reference for information about why the Apply program failed. If problems persist, contact IBM Software Support.

DSYD028I: A synchronization element for database *database name*, table *schema name.table name*, primary key *primary key value*, from device *device id* was rejected due to reject code.

Explanation: The synchronization element for the relational database row specified was not accepted for synchronization for the reason given.

User Response: No action required.

DSYD029W: The DB2 Everyplace Sync Server detected a syntax error in the user WHERE clause of the mirror table *full table name* in database *database name*. Message from parser WHERE clause

Explanation: The DB2 Everyplace Sync Server parses the WHERE clause to obtain the information to handle WHERE clauses that involve multiple tables. If the WHERE clause refers to only one table, this warning can be safely ignored.

User Response: Correct the WHERE clause in the Mobile Devices Administration Center. If problems persist, contact IBM Software Support.

DSYD030I: Usage: com.ibm.mobileservices.adapter.rdb.Replicate mirror_database_name

Explanation: com.ibm.mobileservices.adapter.rdb.Replicate was not executed with one command-line argument.

User Response: Run com.ibm.mobileservices.adapter.rdb.Replicate with a mirror database name as the command-line argument.

DSYD031I: Replication of mirror database *mirror database name* succeeded.

Explanation: Replication for the mirror database was successful.

User Response: No action is required.

DSYD032E: Look-up table *table name* is not found in the subscription *subscription name*, but is mentioned in the mirror filter of domain *table name*.

Explanation: The filter makes reference to a table that does not exist in Subscription.

User Response: Correct the mirror filter in the Mobile Device Administration Center to make sure the look-up table specified in the filter is part of the subscription definition and try the operation again.

DSYD033E: The DB2 adapter received a request for an invalid subscription ID *subscription id* for device type *device type*, device ID *device id*, and user *user name*.

Explanation: The DB2 adapter received a request for an invalid subscription from the specified user and device. The specified subscription ID no longer exists.

User Response: Reset the users device and try again.

DSYD035E: Replication of *mirror database* is blocked: *reason*.

Explanation: The DB2 Everyplace Sync Server cannot ensure that it is safe to perform the replication because it cannot communicate with some synchronization-enabled servers.

User Response: Check the log folder of the Mobile Device Administration Center. Look for DSYD038E to find out which server is unreachable. If the unreachable server is down, mark it as offline. If the server is up, stop it and restart it.

DSYD037E: No table subscription is defined for *mirror database*.

Explanation: No table subscription is defined for the specified mirror database.

User Response: Create a table subscription using the specified mirror database name before requesting replication.

DSYD038E: The last three attempts to communicate with server *server* has resulted in communication failure.

Explanation: The server is either down, unreachable, or there are unknown communication problems.

User Response: Check that the server is up and reachable on the network.

DSYD039E: The data filter *data filter name* for column *column name* in table *table name* is undefined in group *group name* or user *user name* .

Explanation: A required data filter is undefined for an unsubscribed column.

User Response: Ensure that the parameters of vertical filtering are spelled correctly and defined by the group or user. Make corrections in the Subscription, Group, and/or User folders in the Mobile Device Administration Center.

DSYD040E: The migration of DB2 Everyplace is not complete.

Explanation: The migration phase of the configuration of DB2 Everyplace failed.

User Response: Run the DSYconfig utility to complete the migration of DB2 Everyplace. This utility migrates all the DB2 Everyplace Sync Server internal control tables to the current version.

DSYD041E: The mirror filter of *table name* in *subscription name* is invalid. Error: *error*. Filter: *filter*

Explanation: The filter has one of the following problems: 1. The filter is syntactically incorrect. 2. A value for a parameter in the filter is incorrect. 3. The DB2 Everyplace Sync Server cannot identify the tables referenced in the filter.

User Response: Check the filter with the following steps: 1. Check the filter syntax with the DB2 SQL Reference. If the source database is DB2 and all the target table and column names are identical to their corresponding names in the source, you can run the filter on the source to

validate it. 2. Check the default values of the parameters at the group and user levels. 3. If neither of the above checking shows any error, modify the filter by adding a pound sign (#) in front of each table referenced in the filter.

DSYD042E: The following mirror databases have not completed replication: *failed databases*

Explanation: The product cannot be upgraded until all mirror databases have completed replication.

User Response: Replicate the specified mirror databases before starting installation.

DSYD043E: The following DB2 Everyplace Sync Servers are running: *running servers*

Explanation: The product cannot be upgraded until all DB2 Everyplace Sync Servers have been shut down.

User Response: Shut down all DB2 Everyplace Sync Servers before starting installation.

DSYD044E: The following mirror databases have not completed the internal staging operation: *failed databases*

Explanation: The product cannot be upgraded until all mirror databases have completed the internal staging operation.

User Response: Block all subscriptions. Start the DB2 Everyplace Sync Server. Wait until there is no activity. Shut down the DB2 Everyplace Sync Server. Start installation again.

DSYD045E: The following tables must be migrated to a tablespace with a large enough page size: *failed tables*

Explanation: The product cannot be upgraded until the specified tables have been migrated to a tablespace with a large enough page size.

User Response: Please search for DSYD045E in the Installation Notes.

DSYD046W: Before installing DB2 Everyplace version 8.2, all users that have data changes on the device should perform synchronization. Otherwise, if the DB2 Everyplace Sync Client program has not been upgraded to version 8.2, these data changes on the device might be lost in the first synchronization session after this installation.

Explanation: Some users might be forced to perform a refresh by the DB2 Everyplace Sync Server in the first synchronization session after this installation. Only DB2 Everyplace Sync Client version 8.2 or above can perform a refresh without potential data loss.

User Response: If users have important data on the device that is not synchronized, we recommend that you do not install. Ask the users to synchronize their data and then start the installation again.

DSYD047E: Migration failed because *reason.*

Explanation: Migration failed because of the specified reason.

User Response: Correct the problem. Then, run dsymigration.

DSYD048E: Mirror database name is not specified.

Explanation: A mirror database name is required to perform the operation.

User Response: Provide the mirror database name as input to the requested operation.

DSYD049I: The replication process is preparing to replicate the tables in the database *database name.*

Explanation: The replication process is preparing to replicate the tables in the specified database.

User Response: No action is required.

DSYD050I: The replication process is replicating the tables in the database *database name.*

Explanation: The replication process is replicating the tables in the specified database.

User Response: No action is required.

DSYD051I: The replication process is replicating data from the database *database name* to the database *database name*.

Explanation: The replication process is replicating data from the first database specified to the second database specified.

User Response: No action is required.

DSYD052I: The replication process is replicating data from *table name in database name* to *table name in database name*.

Explanation: The replication process is replicating data from the first table specified to the second table specified.

User Response: No action is required.

DSYD053I: The replication process is replicating the row with primary key *primary key value* from *table name in database name* to *table name in database name*.

Explanation: The replication process is replicating the specified row from the first table specified to the second table specified.

User Response: No action is required.

DSYD054I: The replication process is finished replicating *x* rows from *table name in database name* to *table name in database name*.

Explanation: The replication process is finished replicating the specified number of rows from the first table specified to the second table specified.

User Response: No action is required.

DSYD055I: The replication process is pruning the control tables associated with *table name in database name*.

Explanation: The replication process is pruning the control tables associated with the specified table in the specified database.

User Response: No action is required.

DSYD056I: The replication process is finished pruning *x* rows from the control tables associated with *table name in database name*.

Explanation: The replication process is finished pruning the specified number of rows from the control tables associated with the specified table in the specified database.

User Response: No action is required.

DSYD057I: The replication process is finished replicating the tables in the database *database name*.

Explanation: The replication process is finished replicating the tables in the specified database.

User Response: No action is required.

DSYD058I: The replication process is finalizing the replication of the tables in the database *database name*.

Explanation: The replication process is finalizing the replication of the tables in the specified database.

User Response: No action is required.

DSYD059I: The replication process is recovering the previous replication of the database *database name* (*state = state*).

Explanation: The previous replication of the specified database failed. The current replication is performing recovery starting from the specified state.

User Response: No action is required.

DSYF000E: MDSS encountered an exception *exception details.*

Explanation: The MDSS Servlet encountered an exception.

User Response: Gather trace and log files, and contact IBM Software Support for the specific adapter.

DSYG001E: An unexpected exception occurred: *exception.*

Explanation: DB2 Everyplace Sync Server encountered the specified unexpected exception.

User Response: Report the error to your administrator.

DSYG003E: Unable to connect to database *database name.*

Explanation: A connection to the specified database could not be established.

User Response: Verify that the database name, driver, user ID and password is valid and that it is operational.

DSYG004I: Connection to database *database name* **was successful.**

Explanation: A connection to the specified database was successfully established.

User Response: No action required.

DSYG005E: A valid license for this product installation was not found.

Explanation: A valid license must be installed for this product.

User Response: Please contact IBM for information on purchasing this product.

DSYG006I: The Evaluation License for this product installation expires in *remaining days* **days.**

Explanation: This product installation has been licensed for evaluation purposes only, and will cease to operate in the number of days reported.

User Response: Please contact IBM for information on purchasing this product to avoid interruption in service.

DSYG007I: Unable to open log file, exception: *exception.*

Explanation: The DB2 Everyplace Sync Server was unable to open the log file because the specified exception was encountered.

User Response: Verify that the DSYGdflt.properties file, Trace.Path variable specifies a valid path and that the current user has the necessary create/read/write permissions.

DSYG008I: Unable to write to log file *log file name*, **exception:** *exception.*

Explanation: The DB2 Everyplace Sync Server was unable to write to the specified log file because the specified exception was encountered.

User Response: Verify that the DSYGdflt.properties file, Trace.Path variable specifies a valid path and that the current user has the necessary create/read/write permissions.

DSYG010E: The DSYCTLDB control database is incompatible with this DB2 Everyplace Sync Server installation: *level*

Explanation: This DB2 Everyplace Sync Server requires that the installation be at the same level as the DSYCTLDB control database.

User Response: If you are upgrading or installing a new DB2 Everyplace Sync Server, the DSYCTLDB control database needs to be installed or upgraded to the same level using the install package. If the DSYCTLDB database has been upgraded, then this DB2 Everyplace Sync Server installation needs to be upgraded to the same level. If neither of these actions applies, or is unsuccessful in resolving the problem, please contact IBM Software Support.

DSYG011E: Unable to read file *filename*.

Explanation: The specified file could not be read because the file does not exist or the permissions prohibit this action.

User Response: Verify that the specified file exists and that it is accessible.

DSYG013E: Could not load the CustomLogicPolicy subclass *classname*.

Explanation: A class designated by the DB2 Everyplace Administrator to be an implementation of a CustomLogicPolicy could not be loaded.

User Response: Ensure that the specified CustomLogicPolicy subclass is accessible by all SyncServer instances, by any standalone dsyreplicate process, and by the Mobile Devices Administration Center and XML Scripting Tool. If this is unsuccessful in resolving the problem, please contact IBM Software Support.

DSYG014I: Blocking mirror database *database name*.

Explanation: Requesting the servers to stop servicing synchronization and replication requests for this mirror database.

User Response: No action required.

DSYG015I: Updating control database.

Explanation: Applying the requested changes in the control database.

User Response: No action required.

DSYG016I: Unblocking mirror database *database name*.

Explanation: Requesting the servers to continue servicing synchronization and replication requests for this mirror database.

User Response: No action required.

DSYG017E: Cannot edit subscription because there was an error loading the subscription details: *error message*

Explanation: Subscription was not fully initialized earlier and editing at this time is not possible because subscription details are not known.

User Response: Make sure that source and if required mirror databases are accessible and their contents are valid. If there is connection problem, check to make sure URL is correct and drivers are on CLASSPATH. The subscription relies on information in the source and the mirror to fully define itself. When source and/or mirror database is down, actions that can be applied to the subscription are limited.

DSYG018I: Replicating mirror database *database name*.

Explanation: Replicating the mirror database to bring it in sync with the source.

User Response: No action required.

DSYG019I: There are no changes to apply to control database for subscription *subscription name*.

Explanation: No changes will be made in the control database for this operation.

User Response: No action required.

DSYG020I: Creating triggers.

Explanation: This is a progress message.

User Response: No action required.

DSYG021I: Creating replication control tables.

Explanation: This is a progress message.

User Response: No action required.

DSYG022I: Setting up join filtering control structures.

Explanation: This is a progress message.

User Response: No action required.

DSYG023I: Requesting replication.

Explanation: This is a progress message.

User Response: No action required.

DSYG024I: Removing join filtering control structures.

Explanation: This is a progress message.

User Response: No action required.

DSYG025I: Removing replication control tables.

Explanation: This is a progress message.

User Response: No action required.

DSYG026I: Removing triggers.

Explanation: This is a progress message.

User Response: No action required.

DSYG027E: Connection to the database *database name* failed therefore it is not possible to fully initialize the subscription *subscription name*.

Explanation: To be able to retrieve subscription details, access to the specified database is required but the database is either down or there is another connection problem.

User Response: Make sure the specified database is up and running and configured properly. Make sure JDBC Url is correct and JDBC drivers are on Java CLASSPATH.

DSYG028E: Table *table name* is not found in database *database name*.

Explanation: Subscription uses a table that does not exist in the database or the database was not accessible at the time of this error and list of tables could not be retrieved.

User Response: Make sure the specified table exists in the database and verify that the specified database is up and running and configured properly for JDBC access.

DSYG029E: Could not load the tables of subscription *subscription name*. Root cause: *exception message*

Explanation: Details of the subscription's tables were not retrieved due to an error. Actual cause is likely to be a source or a mirror database being down, or a bad database Url or a missing JDBC driver on CLASSPATH.

User Response: Verify that the source and mirror databases are up and running and configured properly for JDBC access. Make sure that the tables which the subscription references still exist.

DSYG030E: DB2 Everyplace Sync Server object *name of missing object* is not defined.

Explanation: An attempt was made to use an undefined object such as a database, subscription, subscription set, group, user or another type that is not yet defined in DB2 Everyplace Sync Server. This object needs to be defined using the Administration console or the XML Scripting tool before it can be used.

User Response: Verify that the object referenced here is actually defined in DB2 Everyplace Sync Server configuration. Cause can be a typo, or misconfiguration.

DSYG031E: DB2 Everyplace Sync Server database *database name* is not defined.

Explanation: An attempt was made to use an undefined mirror or master database. This database needs to be defined using the Administration console or the XML Scripting tool before it can be used.

User Response: Verify that the database referenced here is actually defined in DB2 Everyplace Sync Server configuration. Cause can be a typo, or misconfiguration.

DSYG032E: The DB2 Everyplace Sync Server Context Root is not defined.

Explanation: The DB2 Everyplace Sync Server installation is incomplete as the Context Root is not defined.

User Response: Define a valid context root using command-line configuration tool.

DSYG036E: The executable code retrieved from *code URL* has a build date of *jar build date*, which is inconsistent with the control database build date *db build date*.

Explanation: The DB2 Everyplace Sync Server upgrade installation and configuration has not been completed.

User Response: Ensure that the DB2 Everyplace Sync Server upgrade installation and configuration has been completed for all components. Refer to the DB2e installation documentation for instructions on completing the install configuration.

DSYG037E: Unable to verify the consistency of build dates between DB2e executable code and the control database due to the following error: *exception*.

Explanation: The DB2 Everyplace Sync Server executable code build date or control database build date could not be accessed.

User Response: Ensure that the DB2 Everyplace Sync Server upgrade installation and configuration has been completed for all components. Refer to the DB2e installation documentation for instructions on completing the install configuration.

DSYJ000E: The database adapter detected a non-insertion operation in a Put subscription: *operation*.

Explanation: In a Put subscription, only insertion is allowed. Deletion and update are rejected.

User Response: No action is required.

DSYJ001E: A database error occurred. *the error message from database*

Explanation: A database error occurred. The error message is obtained from the database.

User Response: Refer to the documentation of your database to diagnose the problem. If the problem persists, contact your database administrator.

DSYJ002E: Invalid JDBC driver name *driver name*.

Explanation: The MDSS is unable to determine which database the specified JDBC driver name is using.

User Response: Verify that the JDBC driver name is correct. If it is correct, add the JDBC driver to the file `com\ibm\mobileservices\DSYJdbcDriverList.properties`.

DSYM000E: Unable to start the DB2 Everyplace Mobile Devices Administration Center because the required control database 'DSYCTLDB' does not exist or is invalid.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not be started because the required control database 'DSYCTLDB' does not exist or is invalid.

User Response: Verify the existence of the control database 'DSYCTLDB.' Rerun the script to create the control database (such as `dsyctldb.bat`).

DSYM001E: Unable to locate ID for subscription set *subscription set name*. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not locate the specified subscription set.

User Response: Correct the error and try again.

DSYM006E: Invalid or duplicate subscription set name *subscription set name*. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified subscription set because the name is invalid or another subscription set with the same name exists.

User Response: Specify a unique, valid subscription set name and try again.

DSYM007E: Update subscription set *subscription set name* name failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified subscription set because the name is invalid or it already exists.

User Response: Specify a unique, valid subscription set name and try again.

DSYM008E: Update subscription set *subscription set name* description failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified subscription set description because the description is invalid.

User Response: Specify a valid subscription set description and try again.

DSYM009E: Update subscription set *subscription set name* signature failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified subscription set signature because the signature is invalid.

User Response: Specify a valid subscription set signature and try again.

DSYM010E: Update subscription set *subscription set name* subscriptions failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified subscription set subscriptions. Possible reason: the subscription is invalid or does not exist.

User Response: Verify that the subscription is valid and try again.

DSYM011E: Update subscription set *subscription set name* groups failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified subscription set groups. Possible reason: the group is invalid or does not exist.

User Response: Verify that the group is valid and try again.

DSYM012E: Unable to locate ID for user *user name*. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not locate the specified user.

User Response: Correct the error and try again.

DSYM013E: Invalid or duplicate user name *user name*. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified user because the name is invalid or another user with the same name exists.

User Response: Specify a unique, valid user name and try again.

DSYM014E: Update user *user name* name failed. Action canceled.

Explanation: The DB2 Everywhere Mobile Devices Administration Center could not update the specified user because the name is invalid or it already exists.

User Response: Specify a unique, valid user name and try again.

DSYM015E: Update user *user name* description failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified user description because the description is invalid.

User Response: Specify a valid user description and try again.

DSYM016E: Unable to associate group with user *user name*. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not associate a group with the specified user. Possible reason: the group is invalid or does not exist.

User Response: Verify that the group is valid and try again.

DSYM017E: Update user *user name* password failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified user password because the password is invalid.

User Response: Specify a valid user password and try again.

DSYM018E: Update user *user name* enable state failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified user enable state because the enable state is invalid.

User Response: Specify a valid user enable state and try again.

DSYM019E: Update user *user name* data filters failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified user data filters. Possible reasons: the group level data filter is invalid or does not exist or the user data filter value is invalid.

User Response: Correct the error and try again.

DSYM020E: Unable to remove data filter *data filter name* from the user. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not remove the specified data filter from the user. Possible reasons: the data filter is invalid or is no longer defined at the group level.

User Response: Correct the error and try again.

DSYM021E: Unable to remove data filter *data filter name* from the group. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not remove the specified data filter from the group. Possible reason: the data filter is invalid or no longer exists.

User Response: Correct the error and try again.

DSYM023E: Unable to locate the ID for group *group name*. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not locate the specified group.

User Response: Correct the error and try again.

DSYM024E: Invalid or duplicate group name *group name*. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified group because the name is invalid or another group with the same name exists.

User Response: Specify a unique, valid group name and try again.

DSYM025E: Update group *group name* name failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified group because the name is invalid or it already exists.

User Response: Specify a unique, valid group name and try again.

DSYM026E: Update group *group name* description failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified group description because the description is invalid.

User Response: Specify a valid group description and try again.

DSYM028E: Update group *group name* enable state failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified group enable state because the enable state is invalid.

User Response: Specify a valid group enable state and try again.

DSYM029E: Unable to associate user with group *group name*. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not associate a user with the specified group. Possible reason: the user is invalid or does not exist.

User Response: Verify that the user is valid and try again.

DSYM030E: Unable to associate subscription set with group *group name*. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not associate a subscription set with the specified group. Possible reason: the subscription set is invalid or does not exist.

User Response: Verify that the subscription set is valid and try again.

DSYM031E: Update group *group name* data filters failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified group data filters. Possible reason: the group data filter value is invalid.

User Response: Correct the error and try again.

DSYM032E: Unable to locate ID for subscription *subscription name*. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not locate the specified subscription.

User Response: Correct the error and try again.

DSYM040E: Update file subscription *file subscription name* timestamp failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified file subscription timestamp because the timestamp is invalid.

User Response: Correct the error and try again.

DSYM042E: Update file subscription *file subscription name* source failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified file subscription source because the source is invalid.

User Response: Specify a valid file subscription source and try again.

DSYM043E: Update file subscription *file subscription name* target failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified file subscription target because the target is invalid.

User Response: Specify a valid file subscription target and try again.

DSYM045E: Invalid source database for table subscription *table subscription name*. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center detected an invalid source database for the specified table subscription.

User Response: Verify that the source database is correct and in the {SYSTEM}.{INSTANCE}.{DATABASE} format.

DSYM046E: Invalid or duplicate table subscription name *table subscription name*. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified table subscription because the name is invalid or another table subscription with the same name exists.

User Response: Specify a unique, valid table subscription name and try again.

DSYM047E: Invalid or missing DataPropagator subscription. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not locate the associated DataPropagator subscription.

User Response: Correct the error and try again.

DSYM048E: Unable to create source replication properties for table subscription *table subscription name*. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not create the source replication properties for the specified table subscription.

User Response: Correct the error and try again.

DSYM049E: Unable to create mirror replication properties for table subscription *table subscription name*. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not create the specified table subscription mirror replication properties.

User Response: Correct the error and try again.

DSYM052E: Unable to locate source replication properties id for table subscription *table subscription name*. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not locate the specified table subscription source replication properties.

User Response: Correct the error and try again.

DSYM053E: Unable to locate mirror replication properties id for table subscription *table subscription name*. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not locate the specified table subscription mirror replication properties.

User Response: Correct the error and try again.

DSYM056E: Update table subscription *table subscription name* source system failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified table subscription source system because the source system is invalid.

User Response: Specify a valid table subscription source system and try again.

DSYM057E: Update table subscription *table subscription name* source instance failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified table subscription source instance because the source instance is invalid.

User Response: Specify a valid table subscription source instance and try again.

DSYM058E: Update table subscription *table subscription name* source database failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified table subscription source database because the source database is invalid.

User Response: Specify a valid table subscription source database and try again.

DSYM059E: Update table subscription *table subscription name* source user ID failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified table subscription source user ID because the source user ID is invalid.

User Response: Specify a valid table subscription source user ID and try again.

DSYM060E: Update table subscription *table subscription name* source password failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified table subscription source password because the source password is invalid.

User Response: Specify a valid table subscription source password and try again.

DSYM061E: Update table subscription *table subscription name* mirror database failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified table subscription mirror database because the mirror database is invalid.

User Response: Specify a valid table subscription mirror database and try again.

DSYM062E: Update table subscription *table subscription name* mirror user ID failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified table subscription mirror user ID because the mirror user ID is invalid.

User Response: Specify a valid table subscription mirror user ID and try again.

DSYM063E: Update table subscription *table subscription name* mirror password failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified table subscription mirror password because the mirror password is invalid.

User Response: Specify a valid table subscription mirror password and try again.

DSYM064E: Update table subscription *table subscription name* mirror sync window failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified table subscription mirror sync window because the mirror sync window is invalid.

User Response: Specify a valid table subscription mirror sync window and try again.

DSYM066E: Update table subscription *table subscription name* target database failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified table subscription target database because the target database is invalid.

User Response: Specify a valid table subscription target database and try again.

DSYM067E: Update table subscription *table subscription name* Apply qualifier failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified table subscription Apply qualifier because the Apply qualifier is invalid.

User Response: Specify a valid table subscription Apply qualifier and try again.

DSYM070E: Update table subscription *table subscription name* subtables failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified table subscription subtables because a subtable is invalid.

User Response: Specify a valid table subscription subtable and try again.

DSYM072E: Unable to create DataPropagator table manager control tables for table subscription *table subscription name*. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not complete the changes to the specified subscription due to an exception.

User Response: Correct the error and try again.

DSYM073I: Unable to update DataPropagator table manager control tables for table subscription *table subscription name*.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not complete the changes to the specified subscription due to an exception.

User Response: No action required.

DSYM074E: The replication subscription using source table *source table name* and target table *target table name* does not contain a target column with a primary key. Action canceled.

Explanation: The specified replication subscription table is invalid because it does not contain a target column defined as a primary key.

User Response: Use the advanced subscription definition, dialog, target columns tab to select one or more target columns to be a primary key.

DSYM075E: The DB2 Everyplace Mobile Devices Administration Center encountered an unexpected exception: *exception*. Attempt to correct the error specified in the exception. If unable to do so, close the DB2 Everyplace Mobile Devices Administration Center and try again.

Explanation: The DB2 Everyplace Mobile Devices Administration Center encountered the specified exception. The exception can include additional information that can be used to identify the cause of the error and corrective steps.

User Response: Attempt to correct the error specified in the exception. If unable to do so, close the DB2 Everyplace Mobile Devices Administration Center and try again. If the error persists, contact the database administrator.

DSYM077E: User *user name* password and verify password are not the same. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified user password because the password and verify password are not the same.

User Response: Specify a valid, matching password and verify password and try again.

DSYM079E: Table subscription *table subscription name* source database password and verify password are not the same. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified table subscription source database password because the source database password and source database verify password are not the same.

User Response: Specify a valid, matching source database password and verify password and try again.

DSYM080E: Table subscription *table subscription name* mirror database password and verify password are not the same. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified table subscription mirror database password because the mirror database password and mirror database verify password are not the same.

User Response: Specify a valid, matching mirror database password and verify password and try again.

DSYM082E: Duplicate target table names for table subscription *table subscription name*. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center detected duplicate target table names for the specified table subscription. The target table names within a single table subscription must be unique.

User Response: Specify unique target table names and try again.

DSYM083E: Duplicate target table names for table subscriptions assigned to subscription set *subscription set name*. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center detected duplicate target table names for two or more table subscriptions assigned to the specified subscription set. The target table names of table subscriptions assigned to a subscription set must be unique.

User Response: Ensure that the table subscriptions assigned to the specified subscription set use unique target table names and try again.

DSYM084E: Duplicate target table names for table subscriptions assigned to subscription sets of the group *group name*. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center detected duplicate target table names for two or more table subscriptions in subscription sets assigned to the specified group. The target table names of table subscriptions assigned to subscription sets of a group must be unique.

User Response: Ensure that the table subscriptions assigned to the specified group use unique target table names and try again.

DSYM085E: Assignment of subscription set *subscription set name* to one or more selected groups would result in duplicate table subscription target table names. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center detected that the assignment of the specified subscription set to the group(s) would result in duplicate target table names. The target table names of table subscriptions used by a group must be unique.

User Response: Ensure that the groups using the table subscriptions assigned in the specified subscription set use unique target table names and try again.

DSYM086E: Assignment of the the table subscription *table subscription name* to one or more selected subscription sets would result in duplicate table subscription target table names. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center detected that the assignment of the specified subscription to the subscription set(s) would result in duplicate target table names. The target table names of table subscriptions used by a subscription set must be unique.

User Response: Ensure that the subscription sets using the specified table subscription use unique target table names and try again.

DSYM087E: Assignment of one or more subscription sets to group *group name* would result in duplicate table subscription target table names. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center detected that the assignment of subscription set(s) to the specified group would result in duplicate target table names. The target table names of table subscriptions assigned to subscription sets of a group must be unique.

User Response: Ensure table subscriptions assigned to subscription sets for the specified group contain unique target table names and try again.

DSYM088E: Unable to delete the instance of a DSY default adapter *adapter name*. Action ignored.

Explanation: The DB2 Everyplace Mobile Devices Administration Center detected an attempt to delete the specified DSY default adapter. The specified adapter can not be deleted as it is required by the DB2 Everyplace Mobile Devices Administration Center.

User Response: Do not attempt to remove any DSY default adapter.

DSYM089E: Invalid or duplicate adapter name *adapter name*. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified adapter because the name is invalid or another adapter with the same name exists.

User Response: Specify a unique, valid adapter name and try again.

DSYM090E: Unable to locate ID for adapter *adapter name*. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not locate the specified adapter.

User Response: Correct the error and try again.

DSYM091E: Update adapter *adapter name* name failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified adapter because the name is invalid or it already exists.

User Response: Specify a unique, valid adapter name and try again.

DSYM092E: Update adapter *adapter name* description failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified adapter description because the description is invalid.

User Response: Specify a valid adapter description and try again.

DSYM093E: Update adapter *adapter name* signature failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified adapter signature because the signature is invalid.

User Response: Specify a valid adapter signature and try again.

DSYM095E: Update adapter *adapter name* communication attributes failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified adapter communication attributes because one or more communication attributes are invalid.

User Response: Specify valid communication attributes and try again.

DSYM096E: Update adapter *adapter name* file attributes failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified adapter file attributes because one or more file attributes are invalid.

User Response: Specify valid file attributes and try again.

DSYM098E: Unable to associate adapter with subscription *subscription name*. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not associate an adapter with the specified subscription. Possible reason: The adapter is invalid or does not exist.

User Response: Verify that the adapter exists and is valid and try again.

DSYM099E: Unable to disassociate adapter from subscription *subscription name*. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not disassociate an adapter from the specified subscription. Possible reason: The adapter is invalid or does not exist.

User Response: Verify that the adapter exists and is valid and try again.

DSYM100I: DataPropagator table subscription *table subscription name* created successfully. Additional steps be required before the table subscription can be used for synchronization. Refer to the DB2 Everyplace Sync Server Administration Guide for more information.

Explanation: The DB2 Everyplace Mobile Devices Administration Center successfully created the specified table subscription. However, additional steps be required before the specified subscription can be used for synchronization.

User Response: Refer to the *DB2 Everyplace Sync Server Administration Guide* for further information.

DSYM102E: Unable to delete adapter *adapter name* as one or more subscriptions are currently using it. Action ignored.

Explanation: The DB2 Everyplace Mobile Devices Administration Center detected an attempt to delete the specified adapter. The specified adapter can not be deleted as it is still being used by one or more subscriptions.

User Response: Reassign all subscriptions using the specified adapter to use different adapters and try again (or refresh the adapter object and try again).

DSYM104E: Unable to connect to database *database name* using driver *driver name*, user ID *user ID*.

Explanation: A connection to the specified database could not be established.

User Response: Verify that the database name, driver, user ID and/or password is valid and that it is operational.

DSYM115W: More than *maximum tables* available tables were found in the master database *master database name*. Only the first *maximum tables* tables will be displayed. Use the Filter button to limit the available tables result set.

Explanation: The DB2 Everyplace Mobile Devices Administration Center detected that more than the specified maximum available tables at the specified master.

User Response: If the desired table is not listed, use the Filter button to limit the result set.

DSYM122E: Update custom subscription *custom subscription name* other failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified custom subscription other because the other is invalid.

User Response: Specify a valid custom subscription other and try again.

DSYM124E: Unable to load adapter *adapter name* customizer *customizer class name*. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not load the specified adapter customizer. Possible reason: the adapter communication attributes command is invalid, the class is not found or is not specified in the CLASSPATH environment variable.

User Response: Verify that the the adapter communication attributes command is valid and verify that the command class exists and that it is specified in the CLASSPATH environment variable.

DSYM125E: Unable to load adapter *adapter name* customizer *customizer class name*, exception *exception*. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not load the specified adapter customizer. Possible reason: the adapter communication attributes command is invalid, the class is not found or is not specified in the CLASSPATH environment variable.

User Response: Verify that the the adapter communication attributes command is valid and verify that the command class exists and that it is specified in the CLASSPATH environment variable.

DSYM127E: No valid custom adapters found. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not allow the custom subscription action because no custom (non-DSY) adapters were found.

User Response: Define at least one custom adapter and try again.

DSYM128E: Unable to locate a local database which could be used as a mirror database.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not locate any local database(s) which could be used as a mirror database. At least one local database must be defined for use as a mirror database.

User Response: Ensure at least one local database is defined and try again.

DSYM132E: The AS/400 source database *database name* is not supported for JDBC table subscriptions.

Explanation: The specified database name is not supported by the DB2 Everyplace Sync Server for synchronization using a JDBC table subscription. DB2/400 V5R1 or later is required for DB2 Everyplace Sync Server support for a AS/400 source database.

User Response: Use DB2/400 V5R1 or later for JDBC table subscription support. If using an earlier version of DB2/400, refer to the subscription type DataPropagator table subscription.

DSYM133E: Unable to complete custom subscription *subscription name* as the custom adapter reported a failure while processing the save request. Action canceled.

Explanation: The custom adapter reported that the request to save failed.

User Response: Refer to the documentation for the custom adapter.

DSYM134E: Update adapter *adapter name class name* failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified adapter class name because the class name is invalid.

User Response: Specify a valid adapter class name and try again.

DSYM135E: Duplicate source and mirror databases *database name* found.

Explanation: The table subscription is attempting to use the specified database as both the source and the mirror. The source and mirror databases must be different.

User Response: Specify a different source and mirror database name and try again.

DSYM136E: Update subscription *subscription name* encryption level failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified subscription because the encryption level is invalid.

User Response: Specify a valid encryption level and try again.

DSYM137E: Table *table name* does not contain a primary key. Action canceled.

Explanation: No primary key was found in the specified table. A table must have at least one primary key defined in order to be used as a table subscription.

User Response: Select a table with a primary key and try again.

DSYM138E: Update table subscription *table subscription name* subcolumns failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified table subscription subcolumns because a subcolumn is invalid.

User Response: Specify a valid table subscription subcolumn and try again.

DSYM141E: Invalid or duplicate subscription name *subscription name*. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified subscription because the name is invalid or another subscription with the same name exists.

User Response: Specify a unique, valid subscription name and try again.

DSYM142E: Unable to create subscription *subscription name*. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not create the specified subscription.

User Response: Correct the error and try again.

DSYM143E: Unable to locate ID for subscription *subscription name*. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not locate the specified subscription.

User Response: Correct the error and try again.

DSYM144E: Update subscription *subscription name* name failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified subscription because the name is invalid or it already exists.

User Response: Specify a unique, valid subscription name and try again.

DSYM145E: Update subscription *subscription name* description failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified subscription description because the description is invalid.

User Response: Specify a valid subscription description and try again.

DSYM146E: Update subscription *subscription adapter* adapter failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not update the specified subscription because the adapter is invalid.

User Response: Specify a valid subscription adapter and try again.

DSYM149W: One or more user defined indexes were removed as they will result in duplicate default index(es) using primary keys.

Explanation: The DB2 Everyplace Mobile Devices Administration Center automatically removed one or more user defined indexes as they were duplicates of primary key indexes.

User Response: No action required.

DSYM150I: One or more imported indexes were removed as they would result in duplicate default index(es) using primary keys.

Explanation: The DB2 Everyplace Mobile Devices Administration Center automatically removed one or more imported indexes as they were duplicates of primary key indexes.

User Response: No action required.

DSYM153E: Unable to allow table changes to subscription *subscription name* as the connection to database *database name* using driver *driver name* and user ID *user ID* failed. The define subscription button is disabled.

Explanation: The define subscription button has been disabled because a connection to the specified database could not be established. Typically, this is the result of the password being changed.

User Response: Verify that the database name, driver, user ID and or password is valid and that it is operational by using the test connection button and apply any changes by clicking the OK button. After the changes are applied, try again.

DSYM155E: Unable to use mirror *mirror database name* for source *source database name* as the mirror is already used by another subscription with a different source. Two different source databases not share the same mirror database.

Explanation: The DB2 Everyplace Mobile Devices Administration Center detected that the specified mirror was already in use by another subscription. Source databases not share the same mirror database.

User Response: Change the mirror database name and try again.

DSYM156E: Unable to determine if any DB2 Everyplace devices are installed. One or more DB2 Everyplace device types must be installed in order to create or modify a subscription of this type. Check your DSYIdflt.properties file and ensure that you have a DB2e.InstalledDeviceTypes entry or rerun the DB2 Everyplace installation program.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not determine which DB2 Everyplace devices were installed. One or more DB2 Everyplace device types must be installed in order to create or modify a subscription of this type.

User Response: Check your DSYIdflt.properties file and ensure that you have a DB2e.InstalledDeviceTypes entry or rerun the DB2 Everyplace installation program.

DSYM157E: One or more DB2 Everyplace Sync Server servers were unable to block subscription *subscription name*. Action canceled.

Explanation: One or more DB2 Everyplace Sync Server servers were unable to block the specified subscription. A subscription must be blocked on all alive servers in order to process the edit request.

User Response: The administrator should perform the following steps to ensure that the attempted action will succeed. 1. Ensure that all servers (or the server in a single server environment) are not currently running. 2. Ensure that in the MDAC Server view, all entries on the right pane show that each server is marked as offline. If they are not, right-click on each server and select "Mark as offline". 3. If the servlet engine listens for http requests on a port other than 8080, be sure that the property Server.Port is set to the corresponding value in the DSYGdflt.properties file. After adding this value, delete the server in the MDAC Server view if there is an entry with an incorrect value. Be aware that this port must be accessible from any machine that is running MDAC or MDAC scripting. 4. If more than one server is running on this machine be sure that Server.Name is defined to something unique in the DSYGdflt.properties file. After defining this value, delete the server in the MDAC Server view if there is an entry with an incorrect value. 5. If there is more than one IP address that the server is on, ensure that the one the servlet engine is listening on is defined in DSYGdflt.properties by the Server.IP variable. Again, delete any entries in the MDAC Server view if they contain an incorrect value. 6. To prevent the error in the future, be sure that all servers are shut down by a supported shutdown method, as opposed to killing the process or JVM. In the case of a failure that prevents proper shutdown, open MDAC and mark the servers that are no longer running as offline as describe above. After verifying all of the above, the server be restarted and/or the action be attempted again.

DSYM158E: Unable to connect to the LDAP server *WEBSERVICE_SOAP_ROUTER* using login *user name*. Specify a valid user name and/or password and try again or modify the DSYLDAP.properties to change the LDAP default connection information. LDAP exception: *exception*.

Explanation: Unable to connect to the specified LDAP server using the specified user name and password.

User Response: Specify a valid user name and/or password and try again or modify the DSYLDAP.properties to change the LDAP default connection information.

DSYM159E: Unable to determine the LDAP server to connect to as no *WEBSERVICE_SOAP_ROUTER* was specified in the file *com/ibm/mobileservices/DSYLDAP.properties*. Specify a valid *WEBSERVICE_SOAP_ROUTER* in the *com/ibm/mobileservices/DSYLDAP.properties* file and try again.

Explanation: Unable to determine the LDAP server to connect to as no *WEBSERVICE_SOAP_ROUTER* was specified in the file *com/ibm/mobileservices/DSYLDAP.properties*.

User Response: Specify a valid WEBSERIVCE_SOAP_ROUTER in the com/ibm/mobileservices/DSYLDAP.properties file and try again.

DSYM160W: No groups with DB2e group name prefix name prefix were found in the LDAP server WEBSERVICE_SOAP_ROUTER.

Explanation: No groups with DB2e* prefix (prefix is configurable, it is DB2e by default) were found at the specified LDAP server.

User Response: Define at least one DB2e* group.

DSYM161W: No users were found in the sync group SYNCGROUP, LDAP server WEBSERVICE_SOAP_ROUTER.

Explanation: No users were found in the specified sync group.

User Response: Specify a sync group containing one or more users and try again.

DSYM162W: One or more DB2 Everyplace installed device types do not support table encryption. Table data on these devices will not be encrypted.

Explanation: The DB2 Everyplace Mobile Devices Administration Center has detected that you have requested that the table be encrypted, but one or more DB2 Everyplace installed device types do not support encryption. Thus, table data on these unsupported encryption devices will not be encrypted.

User Response: No action required.

DSYM163E: Table *table name* contains a primary key column with a length exceeding 255 or the total length of all primary key columns exceeds 1024. Action canceled.

Explanation: DB2 Everyplace requires primary key column length not exceed 255 characters. DB2 Everyplace requires that the total length of all primary key columns not exceed 1024 characters.

User Response: Select a valid table and try again.

DSYM165E: One or more DB2 Everyplace Sync Server servers were not offline. All DB2 Everyplace Sync Servers must be marked as offline to create a DataPropagator subscription. Action canceled.

Explanation: One or more DB2 Everyplace Sync Server servers were not offline. All DB2 Everyplace Sync Servers must be marked as offline to create a DataPropagator subscription.

User Response: Mark all servers as offline and try again.

DSYM166W: The DataPropagator subscription *subscription name* is currently read-only. The subscription be viewed, but any changes made will be ignored because one or more DB2 Everyplace Sync Server servers were not offline. All DB2 Everyplace Sync Servers must be marked as offline to edit a DataPropagator subscription.

Explanation: One or more DB2 Everyplace Sync Server servers were not offline. All DB2 Everyplace Sync Servers must be marked as offline to edit a DataPropagator subscription.

User Response: Mark all servers as offline and try again.

DSYM167E: Source table *source table name* has a syntax error in the source to mirror where clause text: *where clause text*. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center detected a syntax error in the source to mirror (all rows needed) where clause in the specified table.

User Response: Validate the source to mirror (all rows needed) where clause for the specified table and try again.

DSYM168I: User *user name*, device id *device id*, device type *device type* was reset.

Explanation: The specified users' device was reset.

User Response: No action required.

DSYM169E: Update subscription *subscription name* custom logic policy type *policy type* failed. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center can not update the specified subscription custom logic policy as the specified policy is invalid.

User Response: Specify a valid policy type and try again.

DSYM170E: One or more column names in table *table name* exceeds *maximum column name length* characters.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not add the specified table as one or more column name lengths exceeded the specified maximum column name length.

User Response: Specify a table with valid column name lengths and try again.

DSYM171E: DB2 Everyplace Mobile Devices Administration Center could not communicate with one or more servers and did not perform replication on the mirror database *mirror database name*, reason: *exception*. Action canceled.

Explanation: DB2 Everyplace Mobile Devices Administration Center could not communicate with one or more servers and did not perform replication on the specified mirror database for the specified exception. To prevent loss of data, the edit request was canceled.

User Response: The administrator should perform the following steps to ensure that the attempted action will succeed. 1. Ensure that all servers (or the server in a single server environment) are not currently running. 2. Ensure that in the MDAC Server view, all entries on the right pane show that each server is marked as offline. If they are not, right-click on each server and select "Mark as offline". 3. If the servlet engine listens for http requests on a port other than 8080, be sure that the property Server.Port is set to the corresponding value in the DSYGdflt.properties file. After adding this value, delete the server in the MDAC Server view if there is an entry with an incorrect value. Be aware that this port must be accessible from any machine that is running MDAC or MDAC scripting. 4. If more than one server is running on this machine be sure that Server.Name is defined to something unique in the DSYGdflt.properties file. After defining this value, delete the server in the MDAC Server view if there is an entry with an incorrect value. 5. If there is more than one IP address that the server is on, ensure that the one the servlet engine is listening on is defined in DSYGdflt.properties by the Server.IP variable. Again, delete any entries in the MDAC Server view if they contain an incorrect value. 6. To prevent the error in the future, be sure that all servers are shut down by a supported shutdown method, as opposed to killing the process or JVM. In the case of a failure that prevents proper shutdown, open MDAC and mark the servers that are no longer running as offline as describe above. After verifying all of the above, the server be restarted and/or the action be attempted again.

DSYM172E: Unable to delete subscription *subscription name*. Action canceled.

Explanation: DB2 Everyplace Mobile Devices Administration Center could not delete the specified subscription due to internal validations failing. To prevent loss of data, the delete request was canceled.

User Response: The administrator should perform the following steps to ensure that the attempted action will succeed. 1. Ensure that all servers (or the server in a single server environment) are not currently running. 2. Ensure that in the MDAC Server view, all entries on the right pane show that each server is marked as offline. If they are not, right-click on each server and select "Mark as offline". 3. To prevent the error in the future, be sure that all servers are shut down by a supported shutdown method, as opposed to killing the process or JVM. In the case of a failure that prevents proper shutdown, open MDAC and mark the servers that are no longer running as offline as describe above. After verifying all of the above, the server be restarted and/or the action be attempted again.

DSYM173E: Unable to delete subscription *subscription name*, reason: *exception*. Action canceled.

Explanation: DB2 Everyplace Mobile Devices Administration Center could not delete the specified subscription due to internal cleanup failing. To prevent loss of data, the delete request was halted.

User Response: The administrator should perform the following steps to ensure that the attempted action will succeed. 1. Ensure that all servers (or the server in a single server environment) are not currently running. 2. Ensure that in the MDAC Server view, all entries on the right pane show that each server is marked as offline. If they are not, right-click on each server and select "Mark as offline". 3. To prevent the error in the future, be sure that all servers are shut down by a supported shutdown method, as opposed to killing the process or JVM. In the case of a failure that prevents proper shutdown, open MDAC and mark the servers that are no longer running as offline as describe above. After verifying all of the above, the server be restarted and/or the action be attempted again.

DSYM174E: Target table name *target table name* already used by another subscription for mirror *mirror database name*. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center detected that the specified target table name was already used by another table subscription for the specified mirror. The target table names within a mirror must be unique.

User Response: Specify a unique target table name and try again.

DSYM175E: Unable to support the Join Filter feature for subscription *subscription name*, target table *target table name*, reason: *reason*. Action canceled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center detected that the specified target table contained a Join Filter that is unsupported for the specified reason.

User Response: Correct the filter for the table and try again.

DSYM176E: Unable to locate a local database which could be used as a source database.

Explanation: The DB2 Everyplace Mobile Devices Administration Center could not locate any local database(s) which could be used as a source database. At least one local database must be defined for use as a source database.

User Response: Ensure at least one local database is defined and try again.

DSYM177I: The source table *table name* contains one or more columns whose datatypes are not supported. The following columns are unavailable for use in the subscription: *column name [datatype name/datatype]*.

Explanation: The DB2 Everyplace Mobile Devices Administration Center detected that the specified source table contained one or more columns that are not supported. The specified columns are unavailable for use in the subscription.

User Response: No action required.

DSYM178I: DB2 Everyplace Mobile Device Administration Center has detected that DB2 UDB Version 8.1 or later is installed. You are now running the DB2 UDB Version 7.2 Control Center which has DB2 Everyplace Sync Server support. Control Center specific objects will be hidden or disabled. Also, DPROPR support will be disabled in the DB2 Everyplace Mobile Device Administration Center. To administrate DPROPR, use the Replication Center and the DB2 Everyplace XML Scripting tool.

Explanation: The DB2 Everyplace Mobile Devices Administration Center detected that DB2 UDB Version 8.1 or later is installed. because DB2 Everyplace Sync Server does not fully support DB2 UDB Version 8.1 or later yet, Control Center specific objects will be hidden or disabled and DPROPR functionality will be disabled in the DB2 Everyplace Mobile Device Administration Center. To administrate DPROPR, use the Replication Center and the DB2 Everyplace XML Scripting tool.

User Response: To administrate your DB2 installation, use the Control Center. To administrate DPROPR, use the Replication Center and the DB2 Everyplace XML Scripting tool.

DSYM179E: Unable to retrieve a list of tables from source database *source database name* using schema pattern *schema pattern*, table pattern *table pattern*. The list of tables to add be empty. Reason: *exception*.

Explanation: The DB2 Everyplace Mobile Device Administration Center was unable to retrieve a list of tables from the specified source database due to the unexpected specified exception.

User Response: Correct the exception and try again.

DSYM180E: The source table *table name* contains one or more columns whose datatypes are not supported and are nullable with no default value specified. The table cannot be used in a table subscription. Action cancelled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center detected that the specified source table contained one or more columns that are not supported and are nullable with no default value specified. The table cannot be used in a table subscription.

User Response: Specify a table that does not contain unsupported datatypes and are not nullable with no default value specified.

DSYM181E: Unable to unblock subscription *subscription name* until a replication on mirror database *mirror database name* is completed. Action cancelled.

Explanation: The DB2 Everyplace Mobile Devices Administration Center detected that the specified subscription could not be unblocked until a replication of the specified mirror database has been completed.

User Response: Perform a replication and try again.

DSYM182W: Unable to support the Join Filter feature for subscription *subscription name*, target table *target table name*, reason: *reason*. Changes were committed.

Explanation: The DB2 Everyplace Mobile Devices Administration Center detected that the specified target table contained a Join Filter that is unsupported for the specified reason. Changes to the subscription were committed.

User Response: Correct the filter for the table and try again.

DSYM183E: Target table *target table name* is invalid. Action canceled.

Explanation: The specified target table name is invalid. Target table names be a maximum of 18 characters in length.

User Response: Specify a valid target table name and try again.

DSYM185I: User *user name*, device id *device id*, device type *device type* was deleted.

Explanation: The specified users' device was deleted.

User Response: No action required.

DSYS000I: Complete synchronization request received; operation type *operation type* operation started for user *user name*, session *session number*, for subscription *subscription name*.

Explanation: A complete synchronization request has been received from the specified user as the last message has been received. The DB2 Everyplace Sync Server will now begin the synchronization process.

User Response: No action required.

DSYS001I: Synchronization ended for user *user name*, session *session number*, for subscription *subscription name*.

Explanation: The synchronization process for the specified user has ended.

User Response: No action required.

DSYS002E: Synchronization failed for user *user name*, session *session number*, for subscription *subscription name*; reason.

Explanation: The synchronization process for the specified user has failed.

User Response: Contact your DB2 Everyplace Sync Server administrator.

DSYS003I: Replication started for database *source database name*.

Explanation: The replication process has begun for the specified database.

User Response: No action required.

DSYS004I: Replication ended for database *source database name*.

Explanation: The replication process has ended for the specified database.

User Response: No action required.

DSYS005E: Replication failed for database *database name: reason*.

Explanation: The replication process for the specified database failed.

User Response: Contact your DB2 Everyplace Sync Server administrator.

DSYS006I: Receiving synchronization request from user *user name*, **session** *session number*, **for subscription** *subscription name*.

Explanation: The specified user initiated the synchronization process and the first message has been received by the DB2 Everyplace Sync Server.

User Response: No action required.

DSYS007I: Restarting synchronization for user *user name*, **session** *session number*, **for subscription** *subscription name*, **from state** *state*.

Explanation: Synchronization was restarted for the specified user by the DB2 Everyplace Sync Server.

User Response: No action required.

DSYS008I: Synchronization started for user *user name*.

Explanation: Synchronization has started for the specified user.

User Response: No action required.

DSYS009I: Synchronization completed for user *user name*.

Explanation: Synchronization has completed for the specified user.

User Response: No action required.

DSYS010I: Starting synchronization.

Explanation: Synchronization is starting for the specified user. No specific subscription has yet been specified.

User Response: No action required.

DSYS011I: Receiving synchronization request for specified subscription.

Explanation: A synchronization request is being received for the specified subscription from the specified user. Server processing on behalf of this request has not yet begun.

User Response: No action required.

DSYS012I: Subscription synchronization in progress.

Explanation: A complete synchronization request has been received for the specified subscription from the specified user, and server processing on behalf of this request has begun.

User Response: No action required.

DSYS013I: Subscription synchronization completed.

Explanation: Server processing of the specified subscription synchronization has completed successfully.

User Response: No action required.

DSYS014E: Subscription synchronization failed.

Explanation: Server processing of the specified subscription synchronization did not complete successfully.

User Response: Contact your administrator.

DSYS015W: Previous session was incomplete, performing session cancellation recovery for user *user name*, session *session number*, for subscription *subscription name*.

Explanation: The messages from a previous subscription synchronization were never retrieved by the client. Session cancellation recovery was needed.

User Response: No action required.

DSYS016I: User *user name* will be forced to refresh table *table name* on its next synchronization request.

Explanation: The DB2 Everyplace Sync Server has determined that the specified User has not performed a synchronization of the specified table for a period of days defined by the property MaxSyncPeriod.Days. On its next synchronization, the User will receive notification that it must refresh the table. Forcing this refresh allows the DB2 Everyplace Sync Server to improve performance for more frequently-synchronizing Users.

User Response: No action required.

DSYS017I: User *user name* upgraded device successfully.

Explanation: The auto deployment servlet has determined that the user has successfully upgraded.

User Response: No action required.

DSYS018E: User *user name* failed to upgrade device.

Explanation: The auto deployment servlet has determined that the user failed to upgrade.

User Response: No action required.

DSYS019I: User *user name* has begun to upgrade device.

Explanation: The auto deployment servlet has determined that the user started the upgrade.

User Response: No action required.

DSYS020E: The directory specified in DSYDeploy.properties does not exist: *directory name*.

Explanation: The directory specified in DSYDeploy.properties does not exist.

User Response: Administration must take action.

DSYS021E: User *security name* is not a member of the required LDAP *sync group* group. Session aborted.

Explanation: The specified user is not a member of the required LDAP group. A user must be a member of the SYNCGROUP specified in com/ibm/mobileservices/DSYLDAP.properties in order to synchronize.

User Response: Contact your administrator.

DSYS023E: User *security name* belongs to more than one *DB2e group name prefix* prefixed LDAP group. Session aborted.

Explanation: The specified user is a member of more than one group that has 'DB2e' prefix (prefix is configurable, it is 'DB2e' by default) which has subscriptions assigned to it. An LDAP user user not belong to multiple DB2e groups due to possible subscription conflicts.

User Response: Contact your administrator.

DSYS024W: The synchronization thread pool has reached maximum capacity.

Explanation: The server has begun to queue new synchronization request because the thread pool is full. This will cause increased response times from the server for new requests.

User Response: If the machine's resources are not fully maximized, then increasing the thread pool size is recommended. The thread pool size is determined based on the property ThreadPoolCount in DSYGdflt.properties.

DSYS025W: The synchronization connection pool has reached maximum capacity.

Explanation: The server has begun to wait for connections because all connections in the connection pool are currently in use. This will cause increased response times from the server for new requests.

User Response: If the machine's database resources are not fully maximized, then increasing the connection pool size is recommended. The connection pool size is determined based on the property Jdbc.MaxConnections in the control database properties table.

DSYS026W: The device *device id* in group *group name* has failed to authenticate because it is not enabled.

Explanation: The device has failed to authenticate because it is not enabled.

User Response: Check to make sure that this device/user/group are enabled.

DSYS027W: The device *device id* for user *user name* has failed to authenticate because it is not associated with a group.

Explanation: The device has failed to authenticate because it is not associated with a group.

User Response: Check to make sure that this user is associated with a group.

DSYS028W: The data sent by user *user name*, device id *device id* cannot be processed. This can happen when the device is deleted or when the schema of a table to which this user subscribes is changed.

Explanation: The device sent data which could not be processed by the server due to an administrative action performed for this user, device, or associated subscription.

User Response: Contact the user and have them re-apply the necessary changes.

DSYS029I: The user *user name* with device id *device id* is synchronizing from an offline refresh image.

Explanation: The user is synchronizing from an offline refresh image.

User Response: No action required.

DSYS030I: The user *user name* with device id *device id* is synchronizing from a backup image.

Explanation: The user is synchronizing from an backup image.

User Response: No action required.

DSYS031E: The backup image for user *user name* with device *device id* cannot be used to synchronize until its previous sessions are processed.

Explanation: A backup image cannot be used until all previous device sessions are complete or the administrator manually deletes device.

User Response: Run the command DSYNSTDIR/Server/bin/dsyrecovery or delete the device.

DSYS032W: The user *user name* with device *device id* cannot synchronize because the subscription is blocked.

Explanation: The user be blocked from synchronizing because the subscription is being edited, was manually blocked by the administrator, replication is currently executing or replication did not finish successfully.

User Response: Unblock the subscription using MDAC. If the subscription is not blocked and no replication is in progress, replicate manually. This will recover any failed replication and make the subscription available for synchronization after it has been edited.

DSYT001E: User *user name* cannot override data filter *data filter name* because the data filter is not defined for this user's group *group name*.

Explanation: Users can only use data filters that are defined in their groups to customize the data filter value based on current user. In this case, the XML script tried to refer to a data filter to specify customized filter value but no such data filter exists in the user's group.

User Response: Fix XML script to reference data filter names that are valid for the related group.

DSYT002E: Data filter *data filter name* is specified more than once.

Explanation: The XML script contains duplicate references for a data filter making the required action unclear.

User Response: Fix XML script to reference data filter names uniquely.

DSYT003E: Subscription *subscription name* is not a jdbc subscription.

Explanation: The XML script tries to change a subscription as if it were a JDBC subscription, where as the subscription type is different.

User Response: Fix XML script to match subscription type you are trying to make changes to.

DSYT004E: Could not create index *index name* on table *table name* in mirror database.

Explanation: The XML scripting tool failed to create the specified index in the mirror database. This be caused by invalid column references, table name or index already exist.

User Response: Fix XML script to specify correct index description.

DSYT005E: Subscription *subscription name* is not an upload subscription.

Explanation: The XML script tries to change a subscription as if it were an Upload subscription, where as the subscription type is different.

User Response: Fix XML script to match subscription type you are trying to make changes to.

DSYT006E: Adapter type does not match subscription type for subscription *subscription name*.

Explanation: The XML script tried to associate a subscription with an adapter where the adapter and subscription types do not match.

User Response: Fix XML script to match the subscription type and adapter type that you are trying to associate the subscription with.

DSYT007E: Subscription *subscription name* is not a file subscription.

Explanation: The XML script tries to change a subscription as if it were a File subscription, where as the subscription type is different.

User Response: Fix XML script to match subscription type you are trying to make changes to.

DSYT008E: Could not unblock subscription *subscription name*.

Explanation: The admin tool could not unblock a subscription that was blocked earlier to prevent synchronization while it was being changed.

User Response: A server that is down be marked as running, thus admin tool is not able to communicate with it, or a running server not be responding. There also be a network problem. Using admin GUI, check to make sure that subscription is not blocked if you wish synchronization to proceed.

DSYT009E: Subscription *subscription name* is not a custom subscription.

Explanation: The XML script tries to change a subscription as if it were a Custom subscription, where as the subscription type is different.

User Response: Fix XML script to match subscription type you are trying to make changes to.

DSYT010E: Conflicting include and exclude tags detected.

Explanation: The XML script had ambiguous tags where it was not clear whether to include or exclude a referenced item.

User Response: Fix XML script to remove ambiguous include or exclude tags.

DSYT011E: "Order" tags used in XML scripts must specify ascending order. XML Scripting tool found order *order index* after processing order *another order index*.

Explanation: The XML script specified incorrect ordering of items such as Subscription Sets in Groups, or Subscriptions in Subscription Sets. When order of items are specified, they must appear in ascending order or the 'order' tag must be omitted in which case the order that the items appear will be used.

User Response: Fix XML script to remove or change ambiguous 'Order' tags.

DSYT012E: Column *column name* with data type *data type name* cannot be replicated. Data type is not supported.

Explanation: The XML script specified a column with an unsupported data type.

User Response: Fix XML script to remove or change columns that have unsupported data types. Depending on the types of databases involved, some data types not be supported. Read documentation about unsupported data types and data type mappings for further information to better design your data synchronization system.

DSYT013E: Subscription *subscription name* has no tables.

Explanation: The XML script did not specify any tables for the table subscription.

User Response: Fix XML script to add tables to the subscription.

DSYT014I: Including table *table name* to subscription *subscription name*.

Explanation: The XML scripting tool is including the table into the subscription to be replicated and synchronized.

User Response: No action required.

DSYT015E: Table *table name* already exists in subscription *subscription name*.

Explanation: Table is already included in the subscription. A table cannot be added to a subscription more than once with the same target schema and table name.

User Response: Fix the XML Script to remove offending table description or change it's target name.

DSYT016E: A table named *table name* does not exist in subscription *subscription name*.

Explanation: XML Scripting tool could not change the details of a table of a subscription because subscription does not have a table with that name.

User Response: Fix the XML Script to remove offending table description or change it's target name to refer to an existing table.

DSYT017I: Altering table *table name* in subscription *subscription name*.

Explanation: XML Scripting tool is about to change the details of a table of a subscription.

User Response: No action required.

DSYT018E: Unexpected join filter value *Y or N or G*.

Explanation: XML Script contains unknown character for join filter setting. It must be one of Y (enabled), N (disabled), or G (Use whatever the Global setting is).

User Response: Fix XML Script to use one of the valid join filter setting identifiers.

DSYT019I: 'JoinFilter' tag for subscription tables does not apply to upload subscriptions. It will be ignored.

Explanation: XML Script contains instruction for join filter setting for a table of an upload subscription where as this setting is used only for replicated subscriptions.

User Response: To avoid this message, fix XML Script to remove join filter settings for tables of upload subscriptions.

DSYT020I: Mirror only index is not applicable in this context. Index *index name* is ignored.

Explanation: XML Script contains instruction to add an index that should be created in the mirror database but subscription does not have a mirror or adding such index is not possible at this time.

User Response: To avoid this message, fix XML Script to remove invalid AddIndex tags.

DSYT021I: Excluding source database column *column name* from target table *table name*.

Explanation: XML Scripting tool is excluding a column of a table. This is typically required if the source database has changed and column no longer exists. Normally you should not exclude any column of a replicated table.

User Response: No action required.

DSYT022I: Setting up synchronization and replication attributes of source column *column name*.

Explanation: XML Scripting tool is processing the setup of a column of a table.

User Response: No action required.

DSYT023E: Source column *column name* is being skipped in the XML script. Column need to have default value for the table replication to succeed. Source column name will be used for target name.

Explanation: The source database table contains a column whose replication and synchronization properties are not specified for the Data DB2 Everyplace Sync Server. Replication process not be able to figure out how to replicate this column. Setup process will continue, however, DB2 Everyplace Sync Server fail trying to use this column.

User Response: Make sure all columns of the source table are listed in the XML Script with correct replication and synchronization settings.

DSYT024I: Blocking subscription *subscription name*.

Explanation: XML Scripting tool is requesting the servers to stop servicing synchronization requests for this subscription.

User Response: No action required.

DSYT025I: Unblocking subscription *subscription name*.

Explanation: XML Scripting tool is requesting the servers to continue servicing synchronization requests for this subscription.

User Response: No action required.

DSYT026E: The value of the element *xml element name* should be a valid integer. Specified value is *value*.

Explanation: XML Scripting tool is expecting an integer, where as it is unable to interpret the specified value as a valid integer.

User Response: Fix the XML content to specify a valid integer where required.

DSYT027E: *SourceTableSpace* tag is not valid when source database is not DB2.

Explanation: It is possible to put a table in a Tablespace in DB2. This tag is used to specify the tablespace of a table in a DB2 database. It is not meaningful if source database is not DB2.

User Response: Fix the XML content to remove this tag.

DSYT028E: Could not block mirror database *database name*.

Explanation: The admin tool could not block a mirror database to prevent synchronization while it was being changed.

User Response: A server that is down be marked as running, thus admin tool is not able to communicate with it, or a running server not be responding, or the mirror database be down. There also be a network problem.

DSYT029E: Cannot change the name of an LDAP user.

Explanation: In LDAP environment, XML Tool cannot edit the name of a user.

User Response: You can try to remove the LDAP user in the LDAP server and add a user with a different name.

DSYT030E: There was an error creating an LDAP user and associating it with DB2 Everyplace Sync Server.

Explanation: XML Tool failed to create a user.

User Response: This be be because the user already exists, or one of it's properties were invalid. There be more information displayed as the root cause.

DSYT031E: There was an error creating an LDAP group and associating it with DB2 Everyplace Sync Server.

Explanation: XML Tool failed to create a group.

User Response: This be be because the group already exists, or one of it's properties were invalid. There be more information displayed as the root cause.

DSYT032E: Cannot change the name of an LDAP group.

Explanation: In LDAP environment, XML Tool cannot edit the name of a group.

User Response: You can try to remove the LDAP group in the LDAP server and add a group with a different security name.

DSYT033E: A valid name for the user must be provided.

Explanation: An invalid name was specified for a user.

User Response: Provide a valid username. In LDAP environment this name must satisfy LDAP requirements also.

DSYT034I: Updating control database.

Explanation: Changes are being written to DB2 Everyplace Sync Server control database.

User Response: No action required.

DSYT035E: A valid name for the group must be provided.

Explanation: An invalid name was specified for a group.

User Response: Provide a valid group name. In LDAP environment this name must satisfy LDAP requirements also.

DSYT036E: *group name* is not a SyncGroup. A valid SyncGroup name that is specified as a SyncGroup in DSYLDAP.properties file must be provided as the SyncGroup of this user.

Explanation: An invalid name was specified for a SyncGroup group. Valid SyncGroups are listed in DSYLDAP.properties file in LDAP enabled DB2 Everyplace Sync Server installations.

User Response: Provide a valid group name that is listed in DSYLDAP.properties as a SyncGroup.

DSYT037E: User's group name *group name* must start with *db2e group name prefix*.

Explanation: In LDAP enabled DB2 Everyplace Sync Server installations, to be able to synchronize, a user needs to belong to 1 'DB2e' group whose name starts with 'DB2e' (or whatever the prefix value is set in configuration properties). This 'DB2e' group is used by DB2 Everyplace Sync Server for synchronization purposes.

User Response: Provide a valid group name that starts with the designated prefix (usually 'DB2e') in addition to other groups this LDAP user already belong to.

DSYT038I: 'WhereClauseMirrorToMobile' and 'WhereClauseMasterToMirror' tags for subscription tables does not apply to upload subscriptions. These tags will be ignored.

Explanation: XML Script contains instruction to add a where clause setting for a table of an upload subscription where as this setting is used only for replicated subscriptions.

User Response: To avoid this message, fix XML Script to remove where clauses for tables of upload subscriptions.

DSYT039E: Create user failed -- possible invalid password length.

Explanation: The creation of the user failed, possibly due to an invalid password length.

User Response: Ensure the specified password conforms to the LDAP security policies as specified by the administrator.

DB2 Everyplace Update Tool error messages

This topic lists all of the error messages that can be generated by the DB2 Everyplace Update Tool.

Table 30 displays the error message and a possible remedy for the problem.

Table 30. Troubleshooting guide

| Error message | Possible remedy |
|---|---|
| Authentication failed (invalid encryption key) - update aborted | Verify that the client settings match the user's settings defined in the Mobile Device Administration Center. |
| File size exceeds available memory | Delete any applications or files that are no longer needed on the device and try again. |
| Internal server error | This is an internal error that you need to report to IBM Software Support with the trace file. |
| Failed to open connection | Check your network connection and the SyncServer. Make sure that the host is connected and the server is running. |
| Failed to establish connection | Check your network connection and the SyncServer. Make sure that the host is connected and the server is running. |
| Failed to send request | Try to synchronize again when there is less traffic on the network or try to synchronize from a faster network. |

Table 30. Troubleshooting guide (continued)

| Error message | Possible remedy |
|---|--|
| Failed to receive reply | Try to synchronize again when there is less traffic on the network or try to synchronize from a faster network. |
| Timeout while receiving reply | Specify a larger timeout value or try to synchronize when there is less traffic on the network. |
| Failed to receive acknowledge | Try to synchronize again when there is less traffic on the network or try to synchronize from a faster network. |
| Failed to open Net library | Verify that the network library exists on the device. Try to reinstall the library. |
| Failed to resolve hostname | Verify that the hostname and the DNS addresses are correct. |
| Failed to allocate working buffer for transport | Delete any applications or files that are no longer needed on the device and try again. |
| Unknown network error | This is an internal error that you need to report to IBM Software Support with the trace file. |
| Failed to create target file | Verify that the target file is not being used by another application. If the target file is being used, unlock it and synchronize again. |
| No files received for update | This is an informational message stating that the server does not have an update for the mobile device. |

DB2eCLP

DB2eCLP is a tool that allows you to directly issue SQL statements from a command-line interface.

This tool is an application that interacts with DB2 Everyplace through a command-line interface. It is used for the DB2 Everyplace mobile database on mobile devices and not used by the Sync Server.

To find this tool, navigate to the following directory: <DSYPATH> \Clients\ *platform* \database\lang\proc\DB2eCLP.

DB2eCLP commands

This topic presents the commands that you can use on DB2eCLP.

You can issue SQL statements directly from a command-line interface. For example:

```
SELECT * FROM PHONEBOOK
```

On some platforms, each statement must end in a semicolon. For example:

```
SELECT * FROM PHONEBOOK;
```

DB2eCLP also supports some extended commands:

\$file [*input file*] [*output file*]

Runs SQL statements from an input file, and writes the result to an output file. This command is not supported on Palm OS systems. For all other platforms, you can specify the full path.

AUTOCOMMIT OFF|ON

Specifies whether the application commits each statement by default (the engine default is ON). When autocommit mode is on (true), each statement is treated as a single, complete transaction. AUTOCOMMIT OFF changes the transaction mode to manual, which enables applications to either roll back or commit work.

BLASTDB

Drops all user tables in the database.

CONNECT TO *arg1*

Automatically disconnects the application from the current connection and reconnects the application to a local database (*arg1*). The specification is in the SQLConnect() CLI call. The delimiter for the paths for CLI-SQLConnect is either \ (backslash) or / (slash). Both delimiters are understood on all platforms and are mapped to the appropriate delimiter when the file system is accessed, which allows databases to reside in different directories. For example,

```
connect to c:\temp\  
create table t (a int)  
insert into t values (10)  
select * from t
```

CONNECT TO *arg1* **USER** *arg2* **USING** *arg3*

Connects the application to a local database (**arg1**) using the specified user name (*arg2*) and password (*arg3*). This information is needed to access encrypted tables. If the application is already connected to another database, that connection is dropped.

A directory name can include a space. For example, C:\System\program files\ is a valid directory structure, but you must use the same directory structure that exists on your machine.

DBCHECK outputfile

Runs the data integrity check tool and writes the result to an output file in the database directory. This command is supported on Linux and Windows 32-bit operating systems only.

DESCRIBE SELECT

Describe the type, column, and name length of the data that is returned by a SELECT statement. For example:

```
DESCRIBE SELECT * FROM PHONEBOOK
```

DISABLE APPLICATION SET DIRTY

Disables setting dirty bit by DB2eCLP.

DISABLE LONG FILENAME

Creates files in 8.3 file name format.

DISABLE PHYSICAL DELETE

Disables physical delete mode (default).

DISABLE READ DELETED

Disables reading deleted rows.

DISABLE REORG

Disables table reorganization.

ENABLE APPLICATION SET DIRTY

Enables setting dirty bit by DB2eCLP.

ENABLE LONG FILENAME

Creates files in long file name format (default).

ENABLE PHYSICAL DELETE

Enables physical delete mode. Deleted rows will no longer be readable.

ENABLE READ DELETED

Enables reading deleted rows.

ENABLE REORG

Enables table reorganization automatically (default).

HELP Lists all available commands.

LIST COLUMNS

List all user table columns in the database.

LIST INDEX

List all indexes ordered by table name, index name, and column order.

LIST TABLES

List all user tables in the database.

VERSION

Prints the DB2 Everyplace mobile database version string. This command returns the same string as the SQLGetInfo() function.

Importing and exporting data using the DB2eCLP

The DB2eCLP for Palm OS, Symbian OS, Windows CE, Windows platforms, Neutrino, and embedded Linux supports importing data from a file to DB2 Everyplace and exporting DB2 Everyplace data to a file.

Importing and exporting data on the Palm OS uses the memo files on the device.

- **To import data from a file on a mobile device to DB2 Everyplace:**

1. Type `IMPORT FROM file_name OF DEL INSERT INTO table_name [(column list)]` where *file_name* is the name of the file to import from.

On Palm OS, *file_name* is the name of the memo to import from. The file name must appear on the first line of the memo. Palm memos have a limitation of storing 4K bytes text. *table_name* is the name of an existing table to import into. For example, to import data from a file named mydata.txt to an existing table named mytable, type:

```
IMPORT FROM mydata.txt OF DEL INSERT INTO mytable
```

- **To export data from DB2 Everyplace to a file:**

1. Type `EXPORT TO file_name OF DEL stmt` where *file_name* is the name of the file to write the data to. *stmt* is the SELECT statement to select the data to export. For example, to export all data from the table named mytable to a file named myfile.txt, type:

```
EXPORT TO myfile.txt OF DEL SELECT * FROM mytable
```

DB2 Everyplace for Palm OS uses a set of command-line tools for Windows and a Palm OS application to import and export data as PDB files. The Import/Export tools include the following executable files, which are installed on the Windows workstation. Download the utilities package of these files from the DB2 Everyplace Support Web page, under Tools and Samples for DB2 Everyplace.

CSV2DB2e.exe

This file imports data from a Comma Separated Values file (with file extension of .csv) into a DB2 Everyplace table. A DB2e table (for example, named PERSON) is represented by two files, DSY_PERSON and DSY_iPERSON. The DSY_PERSON file contains the data, and DSY_iPERSON contains the indexing information.

DB2e2PDB.exe

This program converts a DB2 Everyplace table into Palm OS PDB format. It then copies the files into the user's directory and informs the HotSync program that there are files to install. If more than one user is defined in the system, a user list will be displayed so that you can select the intended user.

PDB2DB2e.exe

PDB2DB2e.exe converts Palm OS PDB files from the user backup area into a DB2 Everyplace table. If more than one user is defined in the system, a user list will be displayed so that you can select the intended user.

DB2e2CSV.exe

DB2e2CSV exports a DB2 Everyplace table into a CSV file. It also makes use of the DB2 Everyplace system catalog files named DB2eSYSTABLES and DB2eSYSCOLUMNS.

PalmImport.bat

PalmImport.bat combines the operations of CSV2DB2e.exe and DB2e2PDB.exe.

PalmExport.bat

PalmExport.bat combines the operations of PDB2DB2e.exe and DB2e2CSV.exe.

The Import/Export tools include the following Palm OS application:

DB2eImport.prc

This program registers the DB2 Everyplace files transferred by the HotSync program to the local DB2 Everyplace system.

- To import data to the Palm OS device:
 1. Create two files with the same name (for example, VNPERSO), one with a .csv extension and one with a .sch extension. The .csv file contains the data, and the .sch file contains the schema for the table to be imported. Note that the file name cannot contain any blank spaces. Sample .csv and .sch files are located in the DemoImport\ folder. The files are named VNPERSO.csv and VNPERSO.sch.

Table 31 lists the supported data types and their representation in the CSV file.

Table 31. Data types and their representation in a CSV file

| Data type | Represented as |
|------------------|------------------------------|
| integer (or int) | 1234 |
| smallint | 1234 |
| decimal(n,p) | 12.34 |
| char(n) | "John" |
| varchar(n) | "John" |
| date | yyyymmdd |
| time | "14.05.48" |
| timestamp | "2001-05-01-16.16.51.000000" |

To represent a null value for a column, enter nothing between the commas of the CSV file. For example, three integer columns with a null in the second column would be represented as 1,,3 in the CSV file and become 1, null, 3 in the database.

2. Start the import tool, PalmImport.bat, providing the schema of the table as a parameter. The schema should be in an associated .sch file. Use the following syntax to start the Import tool:
PalmImport.bat *path_name file_name* where *path_name* is the path to the CSV file, and *file_name* is the CSV file name in uppercase without an extension. The CSV file name cannot be enclosed in double quotation marks or contain any blank spaces. For example:

```
PalmImport.bat DemoImport VNPERSO
```

The imported tables are automatically added to the Palm Install Tool to be installed after the next HotSync operation.

3. Install the DB2eImport.prc to the Palm OS device using the Palm Install Tool.
 4. Perform a HotSync operation to complete the installation of the imported tables and DB2eImport.prc.
 5. Start the DB2eImport program on the Palm OS device to complete the import.
- To export data from the Palm OS device:
 1. Perform a HotSync operation to back up the Palm OS device. Always back up and remove the DB2 Everyplace files (files starting with DSY) from the Palm OS user's backup directory before starting a HotSync operation. The backup area on Windows workstations is usually located at *PalmDir\user_name\Backup* where *PalmDir* is the directory where the Palm OS software is installed, and *user_name* is the user name of the Palm OS user.
 2. Start the export program, PalmExport.bat, with the following syntax:
PalmExport.bat *path_name file_name*

where *path_name* is the output path, and *file_name* is the DB2 Everyplace table name in uppercase. The DB2 Everyplace table name cannot be enclosed in double quotation marks or contain any blank spaces. For example:

```
PalmExport.bat DemoExport VNPERS0N
```

The resulting file is in the same path as the source file.

When an error is encountered, the Import/Export tools report the number of records processed.

A sample using the Import/Export tools is included in the package. The batch files PalmExport.bat and PalmImport.bat provide examples of how to use the CSV2DB2e.exe and DB2e2CSV.exe tools.

Related concepts

“DB2eCLP” on page 238

DB2eCLP is a tool that allows you to directly issue SQL statements from a command-line interface.

Related reference

“DB2eCLP commands” on page 238

This topic presents the commands that you can use on DB2eCLP.

Glossary

Special characters

\$DSYINSTDIR

Refers to the directory where DB2 Everyplace is installed on a Linux or UNIX computer.

<DSYPATH>

Refers to the directory where DB2 Everyplace is installed on a Windows computer.

A

Apply qualifier

A character string that identifies subscription definitions that are unique to each instance of the DataPropagator Apply program.

authentication

The process of validating a user's ID and password against entries in the control database to ensure that the user is authorized to use the DB2 Everyplace Sync Server to synchronize data.

authorization

In computer security, the right granted to a user to communicate with or make use of a computer system.

B

binary large object (BLOB)

A sequence of bytes, where the size of the sequence ranges from 0 to 2 gigabytes. This byte sequence does not have an associated code page and character set. Image, audio, and video objects are stored in BLOBs.

bind In SQL, the process by which the output from the SQL precompiler is converted to a usable structure called an access plan. During this process, access paths to the data are selected and some authorization checking is performed.

BLOB See *binary large object* .

C

client A program or user that communicates with and accesses a database server. You define clients using the Mobile Devices Administration Center.

conflict detection

The process of detecting an out-of-date row in a target table that was updated by a user application. When a conflict is detected, the transaction that caused the conflict is rejected.

Control Center

A graphical interface that shows database objects (such as databases and tables) and their relationship to each other. From the Control Center, you can perform the tasks provided by the DBA Utility, Visual Explain, and Performance Monitor tools.

D

data filter

See *filter*.

data synchronization

See *mobile data synchronization*.

database management system (DBMS)

A computer program that manages data by providing the services of centralized control, data independence, and complex physical structures for efficient access, integrity, recovery, concurrency control, privacy, and security.

database server

A functional unit that provides database services for databases.

DB2 Control Center

See *Control Center*.

DB2 DataPropagator

A replication product that provides an automated method of replicating data from sources to targets. During mobile data synchronization, the mirror and remote databases serve as both source and target. DataPropagator replicates clients' changes from the mirror to the remote database, and also replicates changes from the remote database to the mirror database.

DBCS See *double-byte character set*.

DHCP See *Dynamic Host Configuration Protocol*.

DPROP

See *DB2 DataPropagator*.

double-byte character set (DBCS)

A set of characters in which each character is represented by two bytes.

Dynamic Host Configuration Protocol (DHCP)

An Internet protocol for automating the configuration of computers that use TCP/IP.

E**enterprise database**

See *source database*.

enterprise server

See *source server*.

F

filter A device or program that separates data, signals, or material in accordance with specified criteria.

G

group A collection of clients that have similar mobile data synchronization needs. You define synchronization characteristics for each group, such as which applications the users in the group need to access to perform their jobs and what subsets of enterprise data they need to access.

H**handheld device**

Any computing device that can be held in the hand. Handheld devices include palm-sized PCs and personal digital assistants (PDAs).

I**IBM Sync**

The name for the icon representing the client component of the DB2 Everyplace Sync Server software.

J

join A relational operation that allows for retrieval of data from two or more tables based on matching column values.

K

key A column or an ordered collection of columns that are identified in the description of a table, index, or referential constraint.

L

large object (LOB)

A sequence of bytes, where the length can be up to 2 gigabytes. It can be any of three types: BLOB (binary), CLOB (single-byte character or mixed), or DBCLOB (double-byte character).

LOB See *large object*.

local database

A database that is physically located on the computer in use. Contrast with *remote database*.

log A Mobile Devices Administration Center object containing synchronization error messages and their descriptions.

M

master database

See *source database*.

MDAC

See *Mobile Devices Administration Center*.

mid-tier system

The machine where the DB2 Everyplace Sync Server is installed. In a two-tier synchronization configuration, the mid-tier and source systems refer to the same machine.

mirror database

A database that the DB2 Everyplace Sync Server uses internally to store the data that is required for synchronization and replication.

mobile

Pertaining to computing that is performed on a portable computer or a handheld device by a user who is frequently moving among various locations and using different types of network connections (for example, dial-up, LAN, or wireless).

mobile data synchronization

A two-step process whereby mobile users, or *clients*, submit changes that they made to local copies of source data and receive any changes that were made to source data (in a remote database) since the last time they synchronized.

Mobile Devices Administration Center (MDAC)

A graphical interface that allows you to create, edit, and view synchronization objects and their relationships to each other. The Mobile Devices Administration Center also allows you to view synchronization status of individual clients and error messages.

O

object Anything that can be created or manipulated with SQL, for example, tables, views, indexes, or packages. In object-oriented design or programming, an abstraction consisting of data and operations associated with that data.

ODBC

See *Open Database Connectivity*.

Open Database Connectivity (ODBC)

An API that allows access to database management systems using callable SQL, which does not require the use of an SQL preprocessor. The ODBC architecture allows users to add modules, called database drivers, that link the application to their choice of database management systems at run time. Applications do not need to be linked directly to the modules of all the supported database management systems.

P

PDA See *personal digital assistant* .

persistent

Pertaining to data that is maintained across session boundaries, usually in nonvolatile storage such as a database system or a directory.

personal digital assistant (PDA)

A handheld device that is used for personal organization tasks (such as managing a calendar and note-taking) and includes telephone, fax, and networking features.

pervasive computing (PVC)

The use of a computing infrastructure that includes specialized appliances, known as information appliances, from which users can access a broad range of network-based services (including services that are typically offered through the Internet). These information appliances include televisions, automobiles, telephones, refrigerators, and microwave ovens. Pervasive computing provides convenient access to relevant information and the ability to take action on that information.

primary key

A unique key that is part of the definition of a table. A primary key is the default parent key of a referential constraint definition. With the DB2 Everyplace Sync Server Version 7, each replication source must have one and only one primary key.

privilege

The right to access a specific database object in a specific way. These rights are controlled by users with SYSADM (system administrator) authority or DBADM (database administrator) authority or by creators of objects. Privileges include rights such as creating, deleting, and selecting data from tables.

PVC See *pervasive computing* .

Q

query A request for information from the database based on specific conditions; for example, a request for a list of all customers in a customer table whose balance is greater than \$1000.

R

RAS See *Remote Access Service* .

refresh

A process in which all of the data of interest in a user table is copied to the target table, replacing existing data.

remote database

A database that is physically located on a computer other than the one in use. Contrast with local database. The remote computing device can be stationary and nonportable, or it can be portable.

Remote Access Service (RAS)

A Windows program that manages connections between two systems.

replication

The process of taking changes that are stored in the database log or journal at a source server and applying them to a target server.

replication source

A database table that is defined as a source for replication. After you define a database table as a replication source, the table can accept copy requests.

S

SQL See *Structured Query Language* .

source database

A database residing on a source server containing data to be copied to a target system.

source server

The database location of the replication source.

source table

A table that contains the data that is to be copied to a target table. The source table must be a replication source table. Contrast with *target table*.

subscription

A specification for how the information in a source database is to be replicated to a target database. A subscription allows you to define which subsets of data and files can be copied from the source database. You can create two types of subscriptions: file subscriptions for files stored at the source server and table subscriptions for tables in the source database.

subscription set

A Mobile Devices Administration Center object containing replication subscriptions. To provide group members with access to the data and files defined in replication subscriptions, you create a subscription set and assign subscriptions to it, then assign the subscription set to a group. The subscription set object replaces the application object.

synchronization

See mobile data synchronization.

synchronization object

A manageable item within the Mobile Devices Administration Center that contains information about aspects of the synchronization process in your organization. There are five types of synchronization objects: group, client, subscription set, subscription, and log.

synchronization session

A transaction in which mobile users, or *clients*, submit changes that they made to local copies of source data and receive any changes that were made to source data (residing on the remote server) since the last time they synchronized.

Structured Query Language (SQL)

A programming language that is used to define and manipulate data in a relational database.

T**target database**

A DB2 Everyplace database residing on a mobile device to which data from a source database is copied.

target table

A table to which data from a source table is copied. Mirror tables on the mid-tier server are targets, and DB2 Everyplace tables on the mobile device are targets.

tap To use a stylus to interact with a handheld device.

temporary table

A table created during the processing of an SQL statement to hold intermediate results.

V

view A logical table that consists of data that is generated by a query.

W**wireless LAN**

In wireless uses, a mobile user can connect to a local area network (LAN) through a radio connection. Wireless technologies for LAN connection include speed spectrum, microwave, and infrared light.

Notices

IBM may not offer the products, services, or features discussed in this document in all countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J46A/G4
555 Bailey Avenue

San Jose, CA 95141-1003
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information may contain sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

This product includes software developed by 3Com and its contributors.:

Copyright (c) 1998 3Com/Palm Computing Division. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgement: This product includes software developed by 3Com and its contributors.
4. Neither 3Com nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE 3COM AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL 3COM OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Trademarks

The following terms, which may be denoted by an asterisk (*), are trademarks of International Business Machines Corporation in the United States, other countries, or both.

| | |
|------------------------|-----------|
| AIX | IBM |
| AS/400 | Microsoft |
| Cloudscape | iSeries |
| DataPropagator | OS/390 |
| DB2 | WebSphere |
| DB2 Connect | z/OS |
| DB2 Universal Database | zSeries |
| Everyplace | |

The following terms are trademarks or registered trademarks of other companies:

Intel, Intel Inside (logos), MMX and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names, which may be denoted by a double asterisk (**), may be trademarks or service marks of others.

Index

Special characters

\$DSYINSTDIR vii

<DSYPATH> vii

A

- accessibility ix
- adapters, default 27
- administration
 - device registration 21
 - groups 24
 - subscription sets 55
 - subscriptions 27
 - synchronization objects. See synchronization object 12
 - task road maps 9
 - troubleshooting 157
 - user groups 11, 18
- administration control database. See control database 14
- administrative actions
 - effects on pending client changes 15
- AgentAdapter
 - creating a subscription 46
- Apply program
 - role in synchronization 138, 143
- authentication
 - control database verification 14
 - user passwords 23
- authorization, user. See authentication. 14
- AUTOCOMMIT 238

B

- BLASTDB 239
- books ix

C

- Capture program
 - role in synchronization 138, 143
- characteristics of groups 18
- classpath, adding JDBC drivers to 181
- client
 - configuring synchronization
 - information databases 167, 168
 - database conflicts 157
 - enabling synchronization statistics
 - subscription 166
 - how synchronization happens 136
- Code field
 - error messages 160
- codes for error messages 197
- Command-line scripts for DB2
 - Everyplace Sync Server 175
- configuration
 - Administrator workflow 9
 - configuration, recommending 177

- conflict management 157
- CONNECT TO 239
- contents pane
 - deleting objects 14
 - filtering 17
 - interpreting information 17
 - refreshing 18
- control database
 - description 14
 - importing user definitions 24
- control database, setting properties in 182
- conventions vii
- custom logic
 - API 84
 - data type mapping 83
 - deploying 79
 - developing 77
 - enabling for Visiting Nurse sample 81
 - sample application 82
 - terminology 77
 - uses and limitations 76
 - using for conflict resolution 80

D

- data
 - conflict management 157
 - data filter. See filtering data. 75
 - data integrity check tool 157
 - Data Propagator replication engine 129, 131
 - conflict resolution 129
 - handling changes to the source system 130
 - Data Propagator replication engine versus JDBC replication engine 128
 - data synchronization group. See group 12
 - data synchronization. See synchronization. 11
 - data type mapping 184
 - DB2 186
 - Informix 187
 - Microsoft SQL Server 189
 - Oracle 189
 - restrictions 190
 - data, importing and exporting using DB2eCLP 240
- database
 - administration control information. See control database 24
 - conflict management 157
- Database field
 - error messages 160
- DataPropagator
 - Apply program 138, 143
 - associating a subscription with a subscription set 48
 - Capture program 138
 - DataPropagator (*continued*)
 - configuring for DB2 Everyplace subscriptions 47
 - creating a subscription 48
 - creating a subscription with XML scripting tool 106
 - creating a table subscription 46
 - data source restrictions for subscriptions 54, 191
 - role in synchronization 135
 - DataPropagator (DPROP)
 - Capture program 143
 - DataPropagator subscription
 - defined 28, 46
 - datatype mappings, setting in Sync Server 184
 - DB2 Everyplace
 - information set ix
 - DB2 Everyplace example scenario 3
 - DB2 Everyplace Update Tool
 - error messages 237
 - DB2 Sync
 - client component of Sync Server 137
 - menu options 139
 - settings menu 140, 141
 - subscription sets menu 139
 - DB2 Sync application
 - configuring 138, 139, 149
 - synchronizing data 141, 151
 - DB2 Sync sample application
 - overview 138
 - DB2eCLP 238
 - commands 238
 - importing and exporting data 240
 - DB2eSYSCOLUMNS 195
 - DB2eSYSRELS 196
 - DB2eSYSTABLES 195
 - DB2eSYSUSERS 196
 - definitions (user), importing 24
 - deleting a synchronization object 14
 - DESCRIBE SELECT 239
 - Description field
 - error messages 160
 - description, providing for object
 - file subscription 43
 - subscription set 59
 - device See mobile device 21
 - devices
 - resetting 20
 - diagnosis of problems. See troubleshooting 157
 - diagnosis utility 165
 - disability ix
 - DISABLE APPLICATION SET
 - DIRTY 239
 - DISABLE LONG FILENAME 239
 - DISABLE PHYSICAL DELETE 239
 - DISABLE READ DELETED 239
 - DISABLE REORG 239
 - DSYCTLDB database
 - description 14

DSYID variable (user exits) 162
DSYMSG variable (user exits) 162
DSYUserExits.properties file 162

E

ENABLE APPLICATION SET
DIRTY 239
ENABLE LONG FILENAME 239
ENABLE PHYSICAL DELETE 239
ENABLE READ DELETED 239
ENABLE REORG 239
enabling synchronization 21
encrypting user input
user input, encrypting 180
encryption of user ID and passwords 14
environment variables, setting 181
error log. See log 159
error message
log. See log 13
error messages
codes and explanations 197
DB2 Everyplace Update Tool 237
rejected records 159
user exit interface for 161
viewing 160, 161
exit routines, testing 184
exporting data, using DB2eCLP 240

F

file
changing after creating
subscription 42
file subscription
file changes 42
File subscription
defined 40
file subscriptions
creating 45
filter parameters
horizontal mirror 69, 70, 76
uses and syntax limitations 76
vertical mirror 65, 76
filtering data
contents pane 17
for individual users 71
group 62, 65, 69, 75
horizontal 62
horizontally 66, 67, 68, 69, 70
subscription 62, 64, 66, 68
user 62, 65, 70
vertical 63
vertically 63, 64, 65
filters
defined 62
disabling globally 72
enabling globally 72
filter scope feature 72
disabling on a per-table basis 72
enabling on a per-table basis 72
limitations 73
horizontal mirror 67, 68
horizontal source 66
terminology 73
vertical mirror 63, 64

filters (*continued*)
vertical source 63

G

group
assigning subscription sets 57
creating with the XML Scripting
tool 108
evaluating user needs 18
filtering source data for 75
groups
assigning subscription sets 26
assigning users 23, 26
creating 24, 146
definition 24
deleting 21
disabling synchronization 19
enabling synchronization 26
naming 25
resetting 20, 181

H

handling changes to the client
system 131, 133
handling changes to the source
system 132
HELP 239
Host:Port field
error messages 160

I

ID parameter in user exits 162
importing and exporting data 240
importing data, using the DB2eCLP 240
importing user definitions 24
information
synchronization 166
input queue for synchronization
requests 143

J

JDBC
configuring Sync Server for 30
creating a subscription 144
data sources, accessing using
JDBC 30
grouping subscriptions into
subscription sets 56
JDBC drivers, adding to classpath 181
JDBC replication engine 132, 133
conflict resolution 129
JDBC subscriptions, assigning a
subscription set to 60

K

keyboard shortcuts ix

L

limits 192
LIST COLUMNS 239
LIST INDEX 240
LIST TABLES 240
log files
DB2 Everyplace mobile database 155
log, interpreting 159
logon verification. See authentication. 14
Logs folder 159

M

message, error. See error message 160
mirror database
role in synchronization 143
mirror databases
role in synchronization 137
staging tables 137
mobile device
changing users 21
how synchronization happens 135
registering with Sync Server 21
mobile devices
deleting 21
resetting 181
Mobile Devices Administration Center
Administrator 9
interface 10
managing configurations with XML
files 176
setting synchronization
properties 166
starting 10, 175
working with 9
monitoring client synchronization 165
MSG parameter in user exits 162

N

name, providing for object
file subscription 43
subscription set 59
names
reserved 14

O

object. See synchronization object 12
overview of replication subscriptions and
replication engines 127

P

parameters
synchronization batches 34, 90
passwords, setting 23
problems with synchronization. See
troubleshooting. 157
properties, setting in control
database 182
purging log entries automatically
log and trace file, managing 161

Q

queues for synchronization requests 143

R

recovery utility 165
refresh function (contents pane) 18
registration of device 21
replication engine 127
 Data Propagator 129
 JDBC 132
replication subscription. See file
 subscription. 27
replication, initializing 180
reply to synchronization 142
request for synchronization 136
reserved names 14
reset function for file changes 42
resolution of conflicts 157

S

sample applications
 DB2 Sync
 overview 138
 overview 8
schemas, checking 179
screen readers and magnifiers ix
servlet
 testing 155
settings, DB2 Sync 139
setup for synchronization
 Administrator workflow 9
source
 how synchronization happens 142
source database
 how synchronization happens 135
source databases
 staging of updates to 137
source file, specifying 42
SQL
 limits 192
SQL privileges
 influence on 159
statistics
 synchronization 166
subscription 127
 definition 27
 file 41
 role in synchronization 137
 specifying subscription sets 41
Subscription field
 error messages 160
subscription set
 creating with the XML Scripting
 tool 109
Subscription set
 defined 55
subscription set object
 assigning to group 57
subscription set objects
 definition 55
 naming 55
subscription sets
 assigning subscriptions to 41, 58

subscription sets (*continued*)
 assigning to a Custom
 subscription 59
 assigning to a JDBC subscription 60
 assigning to an upload
 subscription 60
 assigning to group 26
 identifying 58
subscription, specify the encryption
 level 31, 44
subscription, specify the source database
 driver and URL 29
subscription, specifying privileges 28
subscription, specifying source tables 28
subscription, specifying target tables 28
subscriptions
 assigning to a subscription set 58
 blocking 179
 bundling in subscription set 55
 checking schemas 179
 custom 46
 Custom
 assigning a subscription set to 59
 diagnosing 179
 identifying 32, 43
 JDBC 32, 33
 mirror database driver and URL 29
 unblocking 184
 upload 36, 39
 assigning a subscription set to 60
 identifying 40
 source database driver and
 URL 37
Sync Client
 synchronization log 161
Sync Server
 changing devices 21
 command-line scripts 175
 configuring 30
 datatype mappings, setting 184
 device registration 21
 overview 4
 starting the application server 147
 testing the servlet 155
 tracing, enabling and disabling 183
 version, displaying 184
synchronization
 authentication. See authentication. 14
 client statistics table structures 169
 creating a JDBC subscription 144
 customizing for groups 11
 database conflicts 157
 disabling groups and users 19
 enabling trace data upload 166
 explanation of 135
 grouping subscriptions 56
 information 166
 input queue 137
 order 159
 querying client information
 tables 169
 recovering 180
 statistics 166
 timestamp 157
 tutorial 144
 using the sample statistics report 169

synchronization groups
 creating 24
synchronization information 166
synchronization object
 adapter 12
 changes that affect pending client
 changes 15
 deleting 14
 displaying information about 17
 editing 14
 group 12
 log 12
 subscription 12
 subscription set 12
 types 12
 user 12
 where stored 14
synchronization objects
 creating 13
 creating from the object tree 13
 groups 24
 subscription sets 55
 users 19
synchronization reply 142
synchronization request 136
synchronization session 135
synchronization statistics 166
synchronizing data using DB2 Sync 141,
 151
syntax diagram
 how to read vii
system catalog base tables,
 description 194

T

table
 conflict management 157
 version tracking 157
table subscriptions, restrictions for 35
tables
 limits for DB2 Everyplace 192
 overview of DB2 Everyplace 194
 staging 137
 system catalog base, description 194
Timestamp
 error messages 160
timestamps, expanding in trace files 183
trace files
 DB2 Everyplace mobile database 155
trace files. See log and trace file,
 managing 160
tracing
 DB2 Everyplace mobile database 155
tracing level, defining
 log and trace file, managing 160
tracing, enabling and disabling 183
troubleshooting
 error messages and codes 197
troubleshooting problems
 error log, viewing 159
 log on client, viewing 161
 monitoring client
 synchronization 165

U

UNIX

- testing the Sync Server servlet 155
- update conflicts, how to handle 157
- upload subscription, defined 36
- upload subscription, specify the encryption level 39
- upload subscription, specifying privileges 38
- upload subscription, specifying source tables 38
- upload subscription, specifying target tables 38

user

- changing devices 21
 - creating with the XML Scripting tool 108
 - defined 18
 - device registration 21
 - evaluating group needs 18
 - importing 24
- User name field
- error messages 160
- user-exit interface for error handling 161
- user. See user. 18

users

- assigning to group 23, 26
- creating 19, 22
- data and file access, providing 26
- deleting 21
- disabling synchronization 19
- enabling synchronization 26
- naming 23
- passwords, setting 23
- resetting 20, 181

V

- variables, setting 181
- VERSION 240
- version tracking for database records 157
- version, Sync Server 184
- view of objects
 - interpreting contents pane 17
 - refreshing 18
- Visiting Nurse sample application
 - enabling custom logic policies 81

W

WebSphere Application Server

- finding execution status of 182
- launching 182
- stopping 183

Windows

- testing the Sync Server servlet 155
- workflow for synchronization setup 9

X

XML Scripting tool

- exporting XML scripts 86
- overview 85
- parameters 86

XML Scripting tool (*continued*)

- restrictions 110
- XML scripting Tool, running 176



Program Number: 5724-D04

Printed in USA

SC18-9997-00

