IBM DB2 Everyplace

Sync Server Administration Guide

Version 8.1.4

SC18-7186-01

IBM DB2 Everyplace

Sync Server Administration Guide

Version 8.1.4

SC18-7186-01

Note:
Before using this information and the product it supports, read the general information under "Notices" on page 231.

Second Edition (October 2003)

This edition applies to Version 8.1.4 of DB2 Everyplace (product number 5724-D04) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC18-7186-00.

Order publications through your IBM representative or the IBM branch office serving your locality or by calling 1-800-879-2755 in the United States or 1-800-IBM-4YOU in Canada.

Table of contents

Chapter 6. The XML Scripting tool . . 131

Chapter 7. Monitoring client synchronization status 147

Chapter 8. Synchronization problems 151

Part 4. Reference 159

Chapter 9. Sync Server reference information 161

Part 5. Appendixes 229

Notices 231

Glossary 235

Index 239

Contacting IBM 243

Part 1. Understanding synchronization

Chapter 1. DB2 Everyplace synchronization

This chapter provides conceptual information about synchronization using DB2 Everyplace. The topics covered are:

- "Components of the DB2 Everyplace solution"
- "Basic setup for the DB2 Everyplace solution" on page 4
- "The synchronization process" on page 5
- "Synchronization from the mobile device to the source database" on page 5
- "Synchronization from the source database to the mobile device" on page 7
- "Synchronization conflict resolution" on page 8

Components of the DB2 Everyplace solution

The DB2 Everyplace solution for mobile data synchronization includes the following main components:

DB2 Everyplace database engine

Software that runs on the mobile device and allows users to access and modify a local copy of data from a source system. See the *DB2 Everyplace Application Development Guide* for information about the DB2 Everyplace database engine and how to write applications that use the database software.

DB2 Everyplace Sync Server

A client/server program that manages two-way data synchronization between a source and a target database.

The Sync Server acts as an intermediary between the synchronization client software on the mobile device and the DB2 UDB database or the JDBC database on the source server. The Mobile Devices Administration Center, a graphical administration tool for the Sync Server, allows you to define subsets of data and files to be accessed by groups of users. The Sync Server accesses this administration information each time a user requests data synchronization.

DB2 Everyplace Mobile Application Builder

A rapid application development tool that allows you to both build and test DB2 Everyplace applications for mobile devices. Refer to the online help for DB2 Everyplace Mobile Application Builder for more information about this tool.

Figure 1 on page 4 shows how the tools in the solution work together.

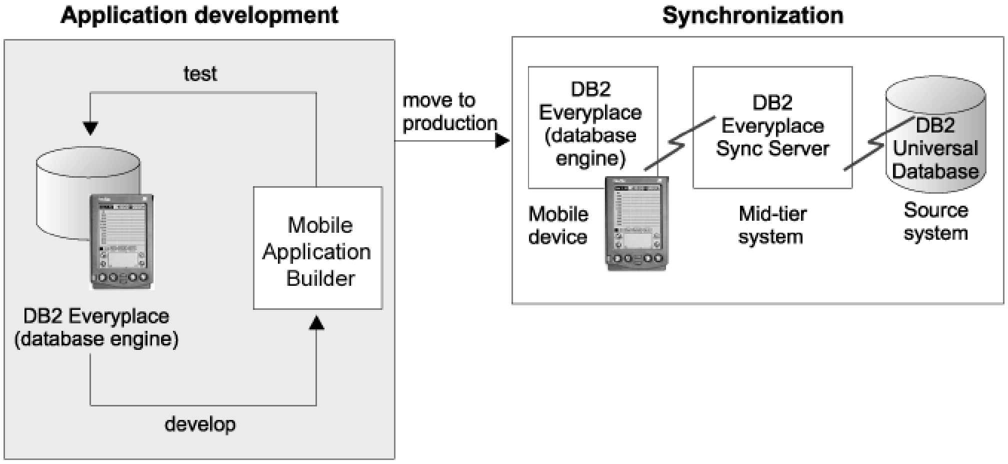


Figure 1. How the tools in the DB2 Everyplace solution work together

Related concepts:

- "Synchronization from the mobile device to the source database" on page 5
- "Synchronization from the source database to the mobile device" on page 7
- "Synchronization conflict resolution" on page 8

Related tasks:

- "Basic setup for the DB2 Everyplace solution"

Basic setup for the DB2 Everyplace solution

Procedure:

Setting up the DB2 Everyplace solution involves the following two tasks:

1. Develop DB2 Everyplace applications.

   These applications provide mobile users access to DB2 Everyplace database tables on their mobile devices. You can develop DB2 Everyplace applications using DB2 Everyplace Mobile Application Builder or another suite of application development tools for mobile devices. See the topics related to application development in the DB2 Everyplace Information Center or PDF version the *DB2 Everyplace Application Development Guide*.
2. Set up synchronization.

   You define data from the source server to be accessed by each group of users. Mobile workers use the DB2 Everyplace Sync Client's API icon to connect to the server and copy source data to their mobile devices. They modify this data using these applications and then synchronize their changes with the source database.

Related concepts:

- "The synchronization process" on page 5
- "Synchronization from the mobile device to the source database" on page 5
- "Synchronization from the source database to the mobile device" on page 7
- "Synchronization conflict resolution" on page 8

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 files and data between the enterprise data source and mobile and embedded devices. 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”
- “Synchronization from the source database to the mobile device” on page 7
- “Synchronization conflict resolution” on page 8
- “Components of the DB2 Everyplace solution” on page 3

Related tasks:

- “Basic setup for the DB2 Everyplace solution” on page 4

Synchronization from the mobile device to the source database

Figure 2 on page 6 shows how changes that a user submits are applied to the source database during synchronization. The numbers in the figure correspond to the explanation following it:

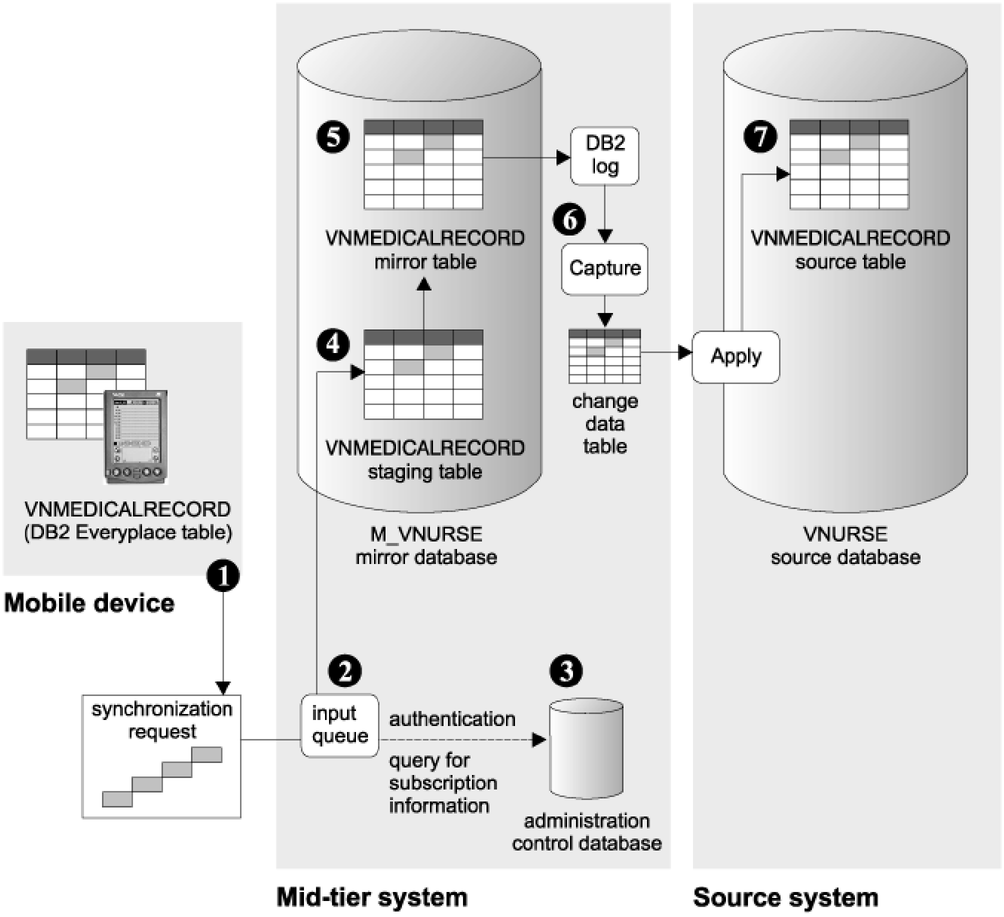


Figure 2. How changes that a user submits for synchronization are applied to the source database

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 **IBM 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 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 7).
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 5
- “Synchronization from the source database to the mobile device”
- “Synchronization conflict resolution” on page 8
- “Components of the DB2 Everyplace solution” on page 3

Related tasks:

- “Basic setup for the DB2 Everyplace solution” on page 4

Synchronization from the source database to the mobile device

Figure 3 shows how changes from a source table are applied to a DB2 Everyplace table on the user’s mobile device during synchronization. The Sync Server sends to the user all relevant source data changes that have been made since the user’s last synchronization. The Sync Server sends only changed data that the user is subscribed to.

The numbers in the figure correspond to the explanations following it.

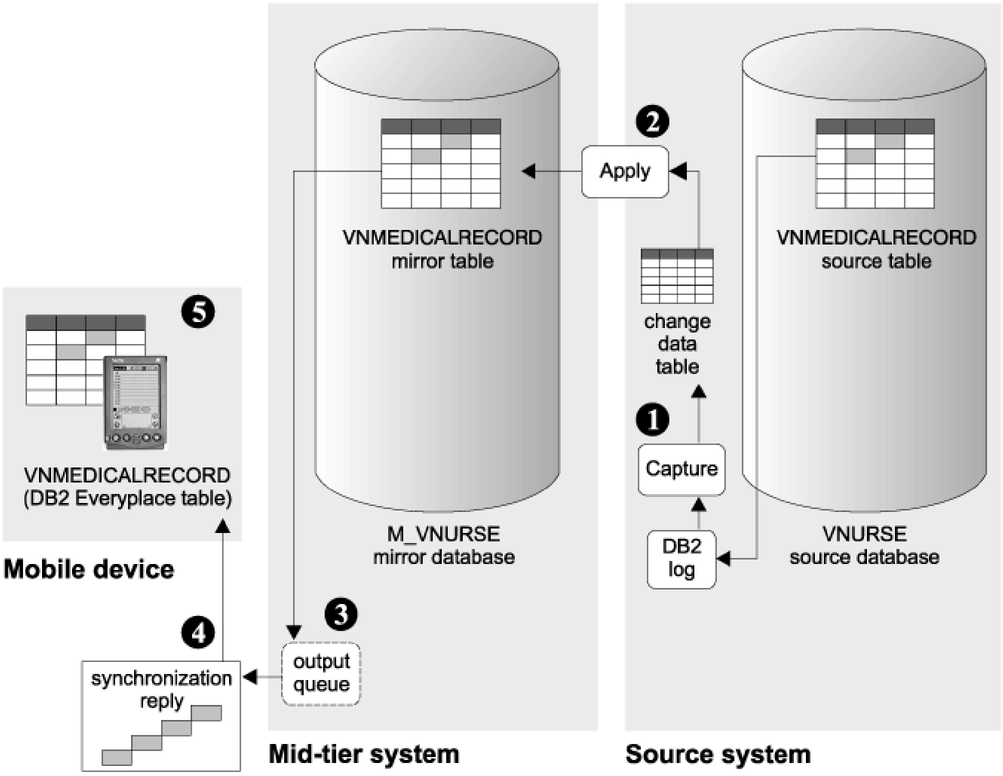


Figure 3. How changes to the source database are applied to the mobile database

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 5
- "Synchronization from the source database to the mobile device" on page 7
- "Synchronization conflict resolution"
- "Components of the DB2 Everyplace solution" on page 3

Related tasks:

- "Basic setup for the DB2 Everyplace solution" on page 4

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 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 Sync Server to determine whether or not 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.

Conflict resolution happens when data is staged to the mirror tables on the mid-tier system, as shown in Figure 4 on page 9. 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.

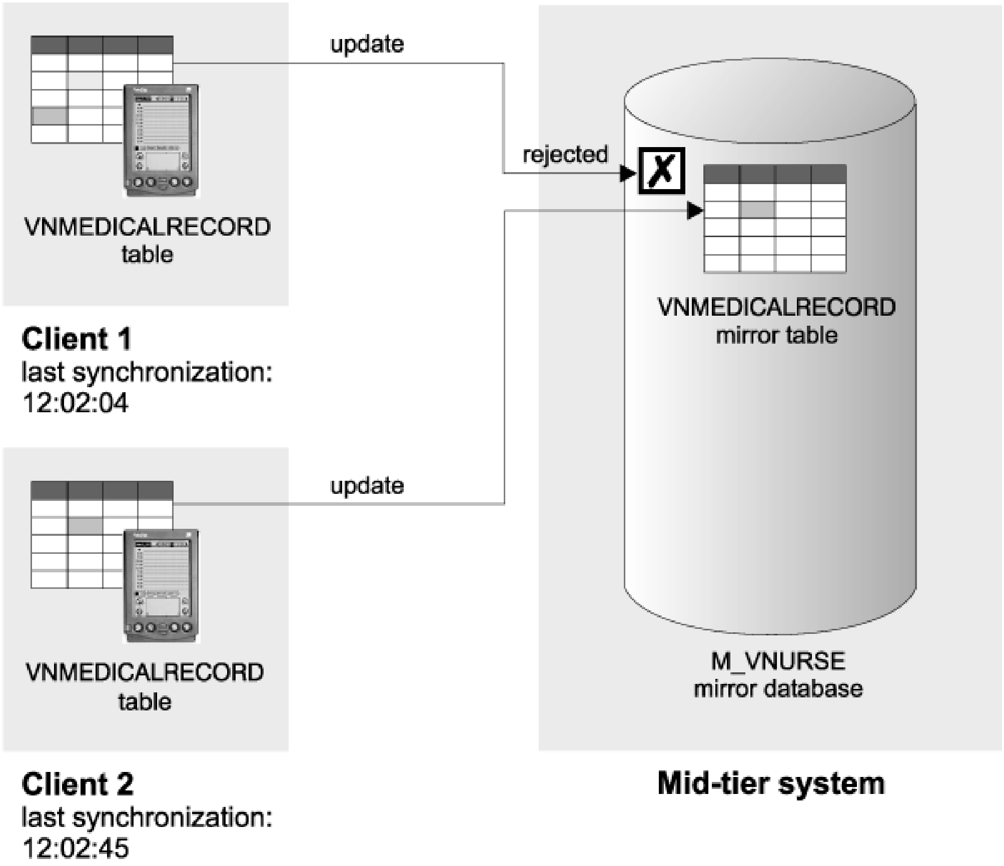


Figure 4. How the Sync Server handles conflicts

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 database.

When DataPropagator applies the changed data from the mid-tier to the source database, additional types of conflicts can occur. See the *DB2 Universal Database Replication Guide and Reference* and the *DB2 Universal Database Administration Guide* for more information on how these conflicts are managed and resolved.

Related concepts:

- "The synchronization process" on page 5
- "Synchronization from the mobile device to the source database" on page 5
- "Synchronization from the source database to the mobile device" on page 7
- "Synchronization conflict resolution" on page 8

Related tasks:

- "Basic setup for the DB2 Everyplace solution" on page 4

Part 2. Configuring synchronization

Chapter 2. Setting up the server

This chapter provides information about how to set up the server for synchronization. The topics covered are:

- “Configuring synchronization”
- “Installing the sample database” on page 14
- “Creating a JDBC subscription to access source data” on page 15
- “Grouping subscriptions into mobile subscription sets” on page 16
- “Creating mobile users” on page 17
- “Creating a data synchronization group” on page 18
- “Starting the application server” on page 19
- “Using a Remote Access Service” on page 20
- “Setting up the serial cradle for use with Windows Remote Access Service” on page 20
- “Configuring the Windows Remote Access Service on the Windows workstation” on page 21

Configuring synchronization

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 tests synchronization using sample data and sample DataPropagator and JDBC subscriptions, but does not use sample upload subscription types. The source tables specified in a DataPropagator subscription must be on a DB2 Universal database. For a JDBC subscription, the source tables can be on any JDBC-compliant database, including DB2 Universal database. For more information on working with different data sources, see “Configuring the Sync Server to access data sources” on page 23.

This tutorial assumes understanding of database administration for DB2 Universal database and JDBC-compliant databases. The following steps are covered:

1. Creating a JDBC subscription to data.
2. Grouping subscriptions into a subscription set.
3. Creating mobile users.
4. Creating a data synchronization group and assigning users and a subscription set to the group.
5. Starting the application server.
6. Using a remote access service.
   - Setting up the serial cradle for use with Windows Remote Access Service
   - Configuring the Windows Remote Access Service on the Windows workstation

Related concepts:

- “Configuring synchronization”

Installing the sample database

Prerequisites:

The DB2 Everyplace Sync Server includes sample databases and configuration for testing JDBC subscriptions and DataPropagator subscriptions. The JDBC sample data can be installed at the end of the installation process. If you installed the JDBC sample data during installation, follow steps 1, 2 and 4. If you did not install sample data during installation, you must install the sample database by following steps 1 to 4.

Procedure:

To install the sample database:

1. Open a DB2 Command Window
2. For Windows, change to the `Server\Sample` directory. For a UNIX operating system, change the directory to `$DSYINSTDIR/Sample`.
3. Install the JDBC sample data.
   - For Windows, run `dsysample.exe`. You will be prompted to provide your DB2 user ID and password, which will be used to create the JDBC sample databases and configuration.
   - For UNIX, run `dsysample.sh db2userid db2password`.
4. Install the DataPropagator sample data:
   - For Windows, run `dsysampleDPropr.bat db2userid db2password`. Click **Install.**
   - For UNIX, run `dsysampleDPropr.sh db2userid db2password`.

   The DB2 user ID and password will be used to create the DataPropagator sample databases.

DB2 Everyplace Sync Server includes a servlet that requires an application server to run. A simple application server is installed by default. DB2 Everyplace Sync Server can be used with many different application servers. For information on how to use the DB2 Everyplace Sync Server with a different application server such as the IBM WebSphere Application Server, see the DB2 Everyplace Web site (`http://www.ibm.com/software/data/db2/everyplace/`).

Related concepts:

- "Configuring synchronization" on page 13

Related tasks:

- "Installing the sample database"
- "Creating a JDBC subscription to access source data" on page 15
- "Grouping subscriptions into mobile subscription sets" on page 16
- "Creating mobile users" on page 17
- "Creating a data synchronization group" on page 18
- "Starting the application server" on page 19
- "Configuring the Windows Remote Access Service on the Windows workstation" on page 21

Creating a JDBC subscription to access source data

The Mobile Devices Administration Center used to manage the Sync Server subscriptions and mobile users.

This task is part of the greater task of Setting up Synchronization. When you have completed the steps below, return to Setting up Synchronization

Procedure:

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 page, 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 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.

   Note: If you testing the connection to a Lotus Domino data source and the connection fails, you must exit the Mobile Devices Administration Center and restart.
10. Click **Close**.
11. On the Mirror page, identify the mirror database that the Sync Server will use to cache data.
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**.
19. Select `DSYSAMPLE.VNSCHEDULE` from the Table list. The **Target schema** and **Target table** fields are automatically completed.

    Note: If you are accessing a Lotus Domino data source, you must enter a schema name in the **Target schema** field. DSYSAMPLE.VNSCHEDULE is an example only. The Target schema field is not automatically complete.
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:
    - Highlight one or more tables and click **Advanced ...**. The Advanced Replication Subscription notebook opens, displaying the highlighted tables.
    - On the Target Columns page, select the checkbox in the **Primary key** column next to the table that you want to have a primary key.
    - Click **OK**. Return to the Define Replication Subscription window.
23. Click the **Timing** push button in the Define Replication Rubscription 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 notebook. 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 concepts:

- "Configuring synchronization" on page 13

Related tasks:

- "Installing the sample database" on page 14
- "Creating a JDBC subscription to access source data" on page 15
- "Grouping subscriptions into mobile subscription sets"
- "Creating mobile users" on page 17
- "Creating a data synchronization group" on page 18
- "Starting the application server" on page 19
- "Configuring the Windows Remote Access Service on the Windows workstation" on page 21

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 Setting up Synchronization

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

Procedure:

To create a subscription set:

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. Select **Create**.
3. Type a name for the subscription set in the **Name** field. Note that the name must match the name of a subscription set on the mobile device. This example uses `VNurse`.
4. Optional: Type a description for the subscription set in the **Description** field. This example uses `Nurse Test Subscription set`.
5. Click the **Subscriptions** tab.
6. Select the subscriptions you created from the **Available subscriptions** list. This example uses `SCHEDULE` and `CONTACT`. Click the **>** push button to move the subscription to the **Selected subscriptions** list.
7. Click **OK** to close the Create Subscription set notebook.

The next step is to create mobile users.

Related concepts:

- "Configuring synchronization" on page 13

Related tasks:

- "Installing the sample database" on page 14
- "Creating a JDBC subscription to access source data" on page 15
- "Creating mobile users"
- "Creating a data synchronization group" on page 18
- "Starting the application server" on page 19
- "Configuring the Windows Remote Access Service on the Windows workstation" on page 21

Creating mobile users

You need to create a user ID and password for mobile users in order to connect the mobile device to the Sync Server. The user ID and password of a mobile user is not a login user ID on the workstation that is running Sync Server. Mobile users can connect to the 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. When you have completed the steps below, return to Setting up Synchronization

Procedure:

To create a mobile user:

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. Leave the **Group** field empty.
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 notebook.

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

Related concepts:

- "Configuring synchronization" on page 13

Related tasks:

- "Installing the sample database" on page 14
- "Creating a JDBC subscription to access source data" on page 15
- "Grouping subscriptions into mobile subscription sets" on page 16
- "Creating a data synchronization group"
- "Starting the application server" on page 19
- "Configuring the Windows Remote Access Service on the Windows workstation" on page 21

Creating a data synchronization group

With DB2 Everyplace Sync Server, you can use groups to manage users, and 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 Setting up Synchronization

Procedure:

To create a group:

1. Right-click on the **Groups** folder. Select **Create**.
2. In the Create Group notebook, 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`. Click the **>** push button to move the user to the **Selected users** list.
6. Click the **Subscription Sets** tab.
7. 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`. Click the **>** push button to move the subscription set to the **Selected subscription sets** list.

8. Complete the Identification, Clients, and Subscription sets pages of the Create Group notebook.
9. Select **Enable Synchronization**.
10. Click **OK** to close the Create Group notebook.

Now you are ready to begin synchronizing data.

Related concepts:

- "Configuring synchronization" on page 13

Related tasks:

- "Installing the sample database" on page 14
- "Creating a JDBC subscription to access source data" on page 15
- "Grouping subscriptions into mobile subscription sets" on page 16
- "Creating mobile users" on page 17
- "Starting the application server"
- "Configuring the Windows Remote Access Service on the Windows workstation" on page 21

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 Setting up Synchronization

Procedure:

If you are using the default application server included with DB2 Everyplace Sync Server, start the server with the following:

- On windows, 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.
- On UNIX, change to `$DSYINSTDIR/server/bin`, then run `dsysync.sh`.

If you are using a different application service with DB2 Everyplace Sync Server, see the DB2 Everyplace Web site at `http://www-3.ibm.com/software/data/db2/everyplace/`.

To verify that the Sync Server servlet is running correctly, go to `http://localhost:8080/db2e/db2erdb` and check for the following message:

```
com.ibm.mobileservices.adapter.rdb.RDb Information
```

Related concepts:

- "Configuring synchronization" on page 13

Related tasks:

- "Installing the sample database" on page 14
- "Creating a JDBC subscription to access source data" on page 15
- "Grouping subscriptions into mobile subscription sets" on page 16
- "Creating mobile users" on page 17
- "Creating a data synchronization group" on page 18
- "Configuring the Windows Remote Access Service on the Windows workstation" on page 21

Using a Remote Access Service

After the DB2 Everyplace Sync Server is running, you must configure a mobile device to communicate with the server. The DB2 Everyplace Sync Server uses TCP/IP to communicate with mobile devices. Most devices support TCP/IP connections using modems, wireless LAN (802.11b), or ethernet interfaces. This task is part of the greater task of Setting up Synchronization. When you have completed the steps below, return to Setting up SyncrhonizationThis section covers the following topics:

- Setting up the serial cradle for use with Windows Remote Access Service
- Configuring the Windows Remote Access Service on the Windows workstation

Related concepts:

- “Configuring synchronization” on page 13

Related tasks:

- “Installing the sample database” on page 14
- “Creating a JDBC subscription to access source data” on page 15
- “Grouping subscriptions into mobile subscription sets” on page 16
- “Creating mobile users” on page 17
- “Creating a data synchronization group” on page 18
- “Starting the application server” on page 19
- “Setting up the serial cradle for use with Windows Remote Access Service”
- “Configuring the Windows Remote Access Service on the Windows workstation” on page 21

Setting up the serial cradle for use with Windows Remote Access Service

This task is part of the greater task of Using a Remote Access Service. When you have completed the steps below, return to Using a Remote Access Service

Prerequisites:

You do not need to complete these steps if you have an ethernet cradle for your Palm OS device.

Alternatively, you can use the MochaSoft Mocha W32 PPP shareware program to connect to the device. For information on Mocha W32 PPP, visit

`http://www.mochasoft.dk/`

Procedure:

To set up the serial cradle for use with Windows Remote Access Service:

1. Connect the serial cradle to the Windows workstation.
2. Open the Control Panel and double-click **Modems**.
3. The Modems Properties window opens. If you have a modem connection currently defined, click **Add** to open the Install New Modem window. If you do not have a modem connection defined, the Install New Modem window displays automatically.
4. In the Install New Modem window, select **Don’t detect my modem**.

5. Click **Next**.
6. Select **Dial-Up Networking Serial Cable between 2 PCs** in the **Models** field.
7. Click **Next**.
8. Choose the serial port the serial cradle is connected to.
9. Click **Next**.
10. Click **Finish** to complete the setup of the Serial cradle for use with Window Remote Access Service.

After you have completed configuring the connection to the device, see the *IBM DB2 Everyplace Installation and User's Guide* to complete the configuration of the Palm OS device using the user ID and password you supplied during this tutorial.

Related concepts:

- "Configuring synchronization" on page 13

Related tasks:

- "Installing the sample database" on page 14
- "Creating a JDBC subscription to access source data" on page 15
- "Grouping subscriptions into mobile subscription sets" on page 16
- "Creating mobile users" on page 17
- "Creating a data synchronization group" on page 18
- "Starting the application server" on page 19
- "Using a Remote Access Service" on page 20
- "Configuring the Windows Remote Access Service on the Windows workstation"

Configuring the Windows Remote Access Service on the Windows workstation

Prerequisites:

You do not need to complete these steps if you have an ethernet cradle for your Palm OS device.

Alternatively, you can use the MochaSoft Mocha W32 PPP shareware program to connect to the device. For information on Mocha W32 PPP, visit `http://www.mochasoft.dk/`

Procedure:

To configure the Windows Remote Access Service on the Windows workstation:

1. Double-click the **Network** icon in the Control Panel.
2. Select the **Services** tab.
3. Remote Access Service should be listed under **Network Services**. If it is not listed. Click **Add**.
4. Select Remote Access Service from the **Network Service** list and Click **OK**.
5. Double-click **Remote Access Service**. The system displays the Remote Access Setup window.
6. Select the Dial-Up Networking Serial Cable between 2 PCs Device.

7. Click **Configure**. The Configure Port Usage window displays, which allows you to specify whether RAS is expecting to dial out on this port, receive connections on this port, or both.
8. For data synchronization, select the **Receive calls only** radio button, then click **OK**.
9. To change the network options, click **Network**. The system displays the Network Configuration window.
10. To ensure that network passwords are received correctly, select the **Allow any authentication including clear text** check box.
11. Select **TCP/IP**. Next to the **TCP/IP** check box, click **Configure**. The system displays the RAS TCP/IP Configuration window. In this window you can set whether the client can access the entire network and how IP addresses are allocated.
12. Select **Entire Network** under Allow remote TCP/IP clients to access.
13. If you have DHCP installed on your network, select **Use DHCP to assign remote TCP/IP client addresses**. If you do not have DHCP installed on your network (or you don not use it), select **Use static address pool**. Set the **Begin** field to a static TCP/IP address provided by the system administrator. Set the **End** field to a static TCP/IP address provided by the system administrator.
14. Click **OK**.
15. Click **OK** to exit the Network Configuration window.
16. Click **Continue** to exit the Remote Access Setup window.
17. Click **Close** to exit the Network window.
18. Reboot the workstation.

    Note: After rebooting the workstation, you must restart the capture process and Sync Server.

After you have completed configuring the connection to the device, see the *IBM DB2 Everyplace Installation and User's Guide* to complete the configuration of the Palm OS device using the user ID and password you supplied during this tutorial.

Related concepts:

- "Configuring synchronization" on page 13

Related tasks:

- "Installing the sample database" on page 14
- "Creating a JDBC subscription to access source data" on page 15
- "Grouping subscriptions into mobile subscription sets" on page 16
- "Creating mobile users" on page 17
- "Creating a data synchronization group" on page 18

Chapter 3. Setting up data sources

This chapter provides information about how to set up data sources for synchronization. The topics covered are:

- "Creating JDBC subscriptions"
- "Creating a DataPropagator table subscription" on page 31
- "Filtering data from data sources" on page 41
- "Creating an application using the remote query and stored procedure adaptor" on page 48

Creating JDBC subscriptions

This chapter provides information about how to set up data sources for synchronization. The topics covered are:

- "Configuring the Sync Server to access data sources"
- "Default JDBC drivers list" on page 25
- "Configuring the Sync Server for JDBC sources using supported drivers" on page 25
- "Configuring the Sync Server for JDBC sources using unsupported drivers" on page 26
- "Specifying the JDBC driver in the Mobile Device Administration Center" on page 27
- "Configuring the JDBC driver to access Lotus Domino data" on page 28
- "Common restrictions for JDBC data sources" on page 29
- "Lotus Domino data source restrictions" on page 30

Configuring the Sync Server to access data sources

This topic provides an overview of the configuration tasks that you perform before you can create a JDBC driver in the Mobile Device Administration Center and synchronize with data on a data source.

Prerequisites:

For JDBC subscriptions the drivers in Table 1 are supported.

Table 1. JDBC supported drivers

Data source	Supported driver
Oracle 8i (8.1.7)	Oracle Thin drivers 8.1.7 classes111.zip
DB2 iSeries Version, V5R1 for OS/400	Toolbox for Java V5R1 **Note:** The DB2 UDB JDBC Driver for AS400 is not supported.
Microsoft SQL Server 2000	SQL Server 2000 Driver for JDBC Service Pack 3a
Sybase Adaptive Server Enterprise, Version 11.93	Sybase jConnect 5.5 for JDBC
Informix Server, Version 9	Informix JDBC driver 1.5
Lotus Domino Server, Version 5.8	Lotus Domino Driver for JDBC, Version 1.1

Table 1. JDBC supported drivers (continued)

Data source	Supported driver
Cloudescape, Version 5.2.1	Cloudscape Network Server

Restrictions:

- For DB2 on iSeries:
  - Due to DB2 on iSeries locking semantics, we support limited concurrent source DELETE operations. If a transaction contains a DELETE operation, the transacation should not rollback if overlapping with replication. If it is rolled back, the subscription will need to be deleted and recreated to restore integrity.
  - The table and row lock wait times of the subscribed tables must be set to a high number for the Sync Server to avoid timing out when waiting for such locks.
- For Sybase:
  - During replication, if a database application changes data in the source database during replication and the changes conflict with replication, the conflicting changes from mobile devices are rejected and the following error message from Sybase is logged:

    ```
    com.sybase.jdbc2.jdbc.SybSQLException: Your server command
     (family id #0, process id #30) encountered a deadlock situation.
      Please re-run your command.
    ```
  - The primary keys of a subscribed table may not be updated in the same statement as the non-primary keys.
- For SQL Server
  - The table and row lock wait times of the subscribed tables must be set to a high number for the Sync Server to avoid timing out when waiting for such locks.
  - The primary keys of a subscribed table may not be updated in the same statement as the non-primary keys.

Procedure:

To configure the Sync Server using supported JDBC drivers:

- add the JDBC driver class path to a Sync Server configuration file,`dsysetjavahome.bat`)
- Lotus Domino only: perform additional configuration steps in “Configuring the JDBC driver to access Lotus Domino data” on page 28
- specify the JDBC driver in the Mobile Device Administration Center

Related tasks:

- “Configuring the Sync Server for JDBC sources using supported drivers” on page 25
- “Specifying the JDBC driver in the Mobile Device Administration Center” on page 27

Related concepts:

- “Default JDBC drivers list” on page 25

Related reference:

- “Lotus Domino data source restrictions” on page 30

Default JDBC drivers list

DB2 Everyplace includes default entries for some JDBC drivers that are already configured and ready to use to access data sources. If you use one of these drivers, you do not need to update the DB2 Everyplace control database before you create a JDBC subscription in the Mobile Device Administration Center. You can view these drivers in the Mobile Device Administration Center on the Create JDBC Subscription window.

Note: DB2 Everyplace dose not bundle these drivers. Make sure that you configure the driver that you are planning to use according to the documentation provided with each driver.

These include:

- For Informix, **com.informix.jdbc.IfxDriver=Ifx**
- For Oracle, **oracle.jdbc.driver.OracleDriver=Oracle**
- For SQL Server,**com.microsoft.jdbc.sqlserver.SQLServerDriver=SQLServer**
- For Sybase, **com.sybase.jdbc.SybDriver=Syb**
- For Lotus Domino, **lotus.jdbc.domino.DominoDriver=Domino**
- For Cloudscape, **com.ibm.db2.jcc.DB2Driver=CS**
- Remote AS\400 database, **com.ibm.as400.access.AS400JDBCDriver=DB2** .

  For AS\400:

  1. Only com.ibm.as400.access.AS400JDBCDriver is supported
  2. DB2 Connect JDBC driver is not supported

If you need to use a different JDBC driver then one of the drivers supported by DB2 Everyplace you must perform an additional step before you can create a subscription in the Mobile Device Administration Center.

Related tasks:

- "Configuring the Sync Server to access data sources" on page 23
- "Configuring the Sync Server for JDBC sources using supported drivers"
- "Specifying the JDBC driver in the Mobile Device Administration Center" on page 27

Configuring the Sync Server for JDBC sources using supported drivers

This task is part of the main task for Configuring the Sync Server for data sources. When you have completed the steps, return to "Configuring the Sync Server to access data sources" on page 23.

For iSeries drivers: Be sure to create a collection library using the SQL statement CREATE SCHEMA.

Procedure:

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 Sync Server servlet or the service if it is running.
3. Using a text editor, open `\Server\bin\dsysetjavahome.bat file` 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\Ora81\jdbc\lib\`, add the following path to the SET JDBC_DRV_CP line of the `dsysetjavahome.bat` file:

   ```
   SET JDBC_DRV_CP=C:\Oracle\Ora81\jdbc\lib\classes111.zip
   ```

   You can specify multiple JDBC client drivers. For example:

   ```
   SET JDBC_DRV_CP=C:\Oracle\Ora81\jdbc\lib\classes111.zip;E:\Informix\JDBC\lib
     \ifxjdbc-g.jar
   ```
5. Adjust the Sync Server properties:
   - Edit the file:

     ```
     \WAS\config\cells\DefaultNode\nodes\DefaultNode\servers\DB2eServer\Server.xml
     ```
   - Add the following line, for example:

     <classpath>C:\Oracle\Ora81\jdbc\lib\classes111.zip<\classpath>
   - You can specify multiple JDBC client drivers.

     ```
     <classpath>C:\Oracle\Ora81\jdbc\lib\classes111.zip<\classpath>
     <classpath>E:\Informix\JDBC\lib\ifxjdbc-g.jar<\classpath>
     ```
6. Restart the Sync Server servlet or the service to make the changes effective.

Related tasks:

- “Configuring the Sync Server for JDBC sources using supported drivers” on page 25
- “Specifying the JDBC driver in the Mobile Device Administration Center” on page 27

Related concepts:

- “Default JDBC drivers list” on page 25

Related reference:

- “Lotus Domino data source restrictions” on page 30

Configuring the Sync Server for JDBC sources using unsupported drivers

This task is part of the main task for Configuring the Sync Server to access data sources. When you have completed the steps, return to “Configuring the Sync Server to access data sources” on page 23.

Procedure:

To configure the Sync Server using unsupported JDCB drivers:

- create a new row in the DB2 Everyplace control database (DSYCTLDB) and provide the appropriate values.
- add the JDBC driver to your system’s class path

To update the `dsy.properties` table in the DB2 Everyplace control database:

1. Exit the Mobile Devices Administration Center.
2. Stop the Sync Server servlet.
3. Create a new row in the DB2 Everyplace control database (table = `dsy.properties`; colums = TYPE, NAME, and VALUE) .
   - For the TYPE column, enter **DSYJdbcDriverList**

- For the NAME column, enter the name of the driver (for example, **com.jnetdirect.jsql.JSQLDriver**)
- For the VALUE column, enter the value for the driver that you are using (for example, **SQLServer**)

A utility is provided, the `dsysetproperty.bat` to query and update the DB2 Everyplace properties table in the DB2 everyplace control database.

- For Windows, you can find `dsysetproperty.bat` located in `%DSYINSTDIR%\Server\bin`.
- For UNIX, `dsysetproperty.sh` is located in `$DSYINSTDIR/Server/bin`.Using a text editor, open `DSYJdbcDriverList.properties` file in `\Server\classes\com\ibm\mobileservices\` directory.

4. Restart the Sync Server servlet to effect the changes.

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 Sync Server servlet if it is running.
3. Using a text editor, open `\Server\bin\dsysetjavahome.bat file` 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\Ora81\jdbc\lib\`, add the following path to the SET JDBC_DRV_CP line of the `dsysetjavahome.bat` file:

```
SET JDBC_DRV_CP=C:\Oracle\Ora81\jdbc\lib\classes111.zip
```

You can specify multiple JDBC client drivers. For example:

```
SET JDBC_DRV_CP=C:\Oracle\Ora81\jdbc\lib\classes111.zip;E:\Informix\JDBC\lib
  \ifxjdbc-g.jar
```

5. Restart the Sync Server servlet to make the changes effective.

Related tasks:

- "Configuring the Sync Server for JDBC sources using supported drivers" on page 25
- "Specifying the JDBC driver in the Mobile Device Administration Center"

Related concepts:

- "Default JDBC drivers list" on page 25

Related reference:

- "Lotus Domino data source restrictions" on page 30

Specifying the JDBC driver in the Mobile Device Administration Center

This task is part of the main task for Configuring the Sync Server for non-DB2 data sources. When you have completed the steps, return to "Configuring the Sync Server to access data sources" on page 23.

Procedure:

In the Mobile Device Administration Center, when you create a JDBC or upload subscription, you must specify the correct driver on the **Source** tab of the Creating JDBC Subscription or Create Upload Subscription notebook.

In most cases, you only need to select the correct source database type from the **Driver** drop-down list. The field to the right is automatically updated with the full name of the JDBC driver.

For DB2 Universal Database, select **DB2 UDB local** if the DB2 database is running on the same machine as the Sync Server. Select **DB2 UDB remote** if the DB2 database is running remotely. For the latter case, make sure the remote DB2 database is configured correctly to allow proper JDBC connection.

Notes:

1. For DB2 Everyplace, select **DB2 UDB local** for both local and remote DB2 databases.
2. For AS400, select **DB2 UDB remote** for the AS400 Toolbox jdbc. **DB2 UDB local** is NOT supported

Related tasks:

- "Configuring the Sync Server for JDBC sources using supported drivers" on page 25
- "Specifying the JDBC driver in the Mobile Device Administration Center" on page 27

Related concepts:

- "Default JDBC drivers list" on page 25

Configuring the JDBC driver to access Lotus Domino data

This task is part of the main task for Configuring the Sync Server for non-DB2 data sources. When you have completed the steps, return to "Configuring the Sync Server to access data sources" on page 23.

Prerequisites:

Be sure that you:

- are using a Lotus Notes Client and Lotus Domino Server 6.0.1 or 6.0.2
- installed the Lotus Notes Client on the same workstation as the Synchronization Server
- perform the configuration steps in "Configuring the Sync Server for JDBC sources using supported drivers" on page 25

Procedure:

To configure Lotus Domino and the DB2 Everyplace Sync Server to use a JDBC driver:

1. Exit the Mobile Device Administration Center if its running.
2. Create a Domino Synchronization Server administration account on the Domino Server (for example, SyncServer Admin) that you want to synchronize with.

   Note: DB2 Everyplace uses the Domino Synchronization Server Administration ID to perform synchronization, as well as to access the Domino database. Therefore, each Domino database that will be used to synchronize must grant a full-control permission to this Domino Synchronization Server Administrator ID.

3. Copy the Notes ID file of the Sync Server Admin account (for example, sadmin.id) that you created in the previous step to the Lotus Notes directory (usually `\Lotus\Notes`) on the workstation where the Synchronization Server is installed.
4. Configure the Notes Client and the Domino Server.
   - On the workstation where the Notes Client is installed:
     a. Ensure that the Notes Client is running with the correct Domino Synchronization Server administration ID.
     b. Copy `dsysetenv_domino.bat` file located in the `Server\Domino` directory to the `Server\bin` directory and rename `dsysetenv_domino.bat` file to `dsysetenv.bat`.
     c. From `Server\Domino`, execute the following command:

        ```
        Instagent appletpath [server]
        ```

        where *appletpath* is the full path of `Server\Domino` and *[server]* is the Domino Server name. For example, **Instagent d:\DB2e\Server\Domino db2www.stl.ibm.com**
   - On the workstation where the Domino Server is installed:
     a. Copy the following files from the Notes Client to the Domino Server:
        - cGetDeltaNative.class
        - getDeltaAgent.class
     b. Modify the Domino Server `notes.ini` file by adding the following line after the Directory entry:

        ```
        JavaUserClasses=Domino_Server-Classpath
        ```

        where *Domino_Server-Classpath* is the directory that `cGetDeltaNative.class` and `getDeltaAgent.class` are located.
5. Restart the Domino Server

Related tasks:

- “Configuring the Sync Server for JDBC sources using supported drivers” on page 25
- “Specifying the JDBC driver in the Mobile Device Administration Center” on page 27

Related concepts:

- “Default JDBC drivers list” on page 25

Related reference:

- “Lotus Domino data source restrictions” on page 30

Common restrictions for JDBC data sources

Common constraints:

- If the structure of a data source table is changed with either an ALTER TABLE or a DELETE TABLE and a CREATE TABLE statement, follow these steps:
  1. Remove the table from all subscriptions.
  2. Execute the ALTER TABLE or the DELETE TABLE and the CREATE TABLE statement.
  3. Add the table back to the subscriptions.

The reason for these steps is to ensure that the source capture triggers remain operable. For example, DB2 renders triggers inoperable upon the execution of any ALTER TABLE or DELETE TABLE statement.

Referential-integrity (RI) constraints:

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

- Do not update the primary keys on the client database or mobile device.
- Do not subscribe tables that have parent-child relationships which involve cycles (for example self-loops).
- Tables must be added in parent-to-child order during subscription creation.
- RI relationships cannot cross any subscription boundaries.
- Do not subscribe tables that have triggers.

Lotus Domino data source restrictions

Domino data source restriction:

- At least one primary key must be selected on each table for Lotus Domino data sources. When you a select primary key use the following rules:
  - The same column must not be identified more than once.
  - The number of identified columns must not exceed 16 and the sum of their length attributes must not exceed 1024
  - No LONG VARCHAR column may be used as part of a primary key
  - Only one primary key can be defined on a table
- Reserved keywords:

  Domino treats the following names as reserved (do not use them to name your columns):
  - NoteID
  - NoteRef
  - NoteCreated
  - NoteModified
  - NoteFile
  - NoteForm
  - NoteFormType
  - NoteUNID
  - NoteRefUNI
  - NoteConflict
  - NoteAccessed
- The number of characters that are returned from Domino text fields:
  - By default, the maximum number of bytes that is allowed in a string associated with a text field is 254, and a rich text field is 512.
  - To get a larger number, maximum is 15360, include these keywords in your URL:

    ```
    jdbc:domino:/<filename>[/Server>][;<keyword>=<attribute>]
    ```

    where <*keyword*> is the MaxVarcharLen for the text field, and the MaxLongVarcharLen for rich text field. For example, **jdbc:domino:/Sample.nsf/angkorwat.svl.ibm.com;MaxVarcharLen=500**.

Related tasks:

- “Configuring the Sync Server for JDBC sources using supported drivers” on page 25
- “Specifying the JDBC driver in the Mobile Device Administration Center” on page 27

Related concepts:

- “Default JDBC drivers list” on page 25

Creating a DataPropagator table subscription

This chapter provides information about how to create a DataPropagator table subscription. The topics covered are:

- “What is a DataPropagator subscription?”
- “What is a replication source?” on page 32
- “Creating a DataPropagator Version 8 table subscription” on page 32
- “Configuring DataPropagator for DB2 Everyplace subscriptions” on page 33
- “Associating the DataPropagator Subscription Set with a DB2 Everyplace table subscription” on page 34
- “Data source restrictions for DataPropagator subscriptions” on page 39
- “Improving replication performance” on page 39
- “Batching synchronization requests” on page 40
- “Replicating on demand” on page 40

What is a DataPropagator subscription?

A *DataPropagator subscription* provides specifications for how data in a source table is to be synchronized to a table on the target, or mobile system. For example, the DataPropagator 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.

Related concepts:

- “What is a replication source?” on page 32
- “Data filtering and replication” on page 111
- “Improving replication performance” on page 39

Related tasks:

- “Using the Mobile Devices Administration Center” on page 77
- “Identifying the subscription” on page 105
- “Specifying the encryption level” on page 106
- “Mapping the source to the target” on page 107
- “Identifying the mirror database” on page 108
- “Adding a replication source” on page 108
- “Creating a filter for replicated data” on page 109
- “Batching synchronization requests” on page 40
- “Providing a user ID and password for the subscription” on page 111

What is a replication source?

A *replication source* is a table residing on a source system from which data is replicated to a target system. You must define at least one replication source for every DataPropagator subscription. You must 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 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

Related concepts:

- "Data filtering and replication" on page 111
- "Improving replication performance" on page 39

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Creating a DataPropagator subscription" on page 104
- "Identifying the subscription" on page 105
- "Specifying the encryption level" on page 106
- "Mapping the source to the target" on page 107
- "Identifying the mirror database" on page 108
- "Adding a replication source" on page 108
- "Creating a filter for replicated data" on page 109
- "Batching synchronization requests" on page 40
- "Providing a user ID and password for the subscription" on page 111

Creating a DataPropagator Version 8 table subscription

Currently, you cannot create or edit a DataPropagator Version 8 table subscription in the Mobile Devices Administration Center. For earlier versions of DataPropagator, you can perform these task in the Mobile Devices Administration Center.

Creation of a DB2 Everyplace Data Propagator Table Subscription will require:

1. In the DB2 UDB Replication Center, setting up the DataPropagator Subscription Set for DB2 Everyplace DataPropagator subscriptions.
2. Utilizing the DB2 Everyplace XML Scripting Tool to associate the Replication Center Data Propagator Subscription Set with a DB2 Everyplace Data Propagator table subscription.

Related tasks:

- "Configuring DataPropagator for DB2 Everyplace subscriptions" on page 33
- "Associating the DataPropagator Subscription Set with a DB2 Everyplace table subscription" on page 34

Related reference:

- "Data source restrictions for DataPropagator subscriptions" on page 39

Configuring DataPropagator for DB2 Everyplace subscriptions

The following sequence of steps will specify the requirements of DB2 Everyplace on the Data Propagator components, and expects that the reader reference the DataPropagator Guide and Reference to ensure that the components are created correctly.

Prerequisites:

Ensure that the source and mirror databases exist and are cataloged on the machine(s) which will be running the DB2 UDB Replication Center and the DB2 Everyplace Mobile Devices Administration Center.

Procedure:

1. Open the DB2 UDB 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. Enable both the source and mirror Capture Control Server databases for replication. If a database does not already have **LOGRETAIN=RECOVERY**, then the Backup Wizard will open, and a backup should be initiated.
5. 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 or not 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.
6. To 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 Data Propagator 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.

DB2 Everyplace will initiate the full refresh during the initial replication.

Related tasks:

- "Creating a DataPropagator Version 8 table subscription" on page 32
- "Associating the DataPropagator Subscription Set with a DB2 Everyplace table subscription"

Related reference:

- "Data source restrictions for DataPropagator subscriptions" on page 39

Associating the DataPropagator Subscription Set with a DB2 Everyplace table subscription

Restrictions:

- By default, DataPropagator will create the mirror tables in their own, non-system managed, tablespaces. If you want the DB2 Everyplace control tables associated with the mirror tables to be created in the same tablespaces, you will need to either override the tablespace creation to specify that the tablespace is system-managed, or increase the tablespace container size to accomodate 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 tablespace in the XML to generate the DB2 Everyplace Subscription, the default tablespace USERSPACE1 will be 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.

The following is an example of an XML Script that you can use to associate a DB2 Everyplace DataPropagator Table Subscription with the previously created Data Propagator Subscription Set using the DB2 Everyplace XML Scripting Tool.

Note: The tablespaces 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 tablespace creation must be modified.

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

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE Root SYSTEM "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>0</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>
   <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>VNPERSON</TargetTable>
   <TableSpace>TSVNPERSON</TableSpace>
   <SourceSchema>DSYSAMPLE</SourceSchema>
   <SourceTable>VNPERSON</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 Version 8 table subscription" on page 32
- "Associating the DataPropagator Subscription Set with a DB2 Everyplace table subscription" on page 34

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:

- Do not create a DataPropagator subscription on tables that have referential-integrity constraints or triggers. Otherwise, replication would fail and would be unrecoverable.
- For z/OS and iSeries source systems, you must issue a bind Capture job to start the Capture program on the source system. For more information on how to bind the Capture program, see the Replication topics in the DB2 Universal Database Information Center.
- The mirror database must be located on the same server as the replication engine. 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, tablespaces. If you want the DB2 Everyplace control tables associated with the mirror tables to be created in the same tablespaces, you will need to either override the tablespace creation to specify that the tablespace is system-managed, or increase the tablespace container size to accomodate 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 tablespace in the XML to generate the DB2 Everyplace Subscription, the default tablespace USERSPACE1 will be 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.

Related tasks:

- "Creating a DataPropagator Version 8 table subscription" on page 32
- "Associating the DataPropagator Subscription Set with a DB2 Everyplace table subscription" on page 34

Improving replication performance

For performance reasons, you probably do not want the Sync Server to perform replication for every synchronization request. To improve performance, you can do one of the following:

- Batching synchronization requests
- Replicating on demand

Related concepts:

- "What is a replication source?" on page 32
- "Data filtering and replication" on page 111

Related tasks:

- "Configuring DataPropagator for DB2 Everyplace subscriptions" on page 33
- "Associating the DataPropagator Subscription Set with a DB2 Everyplace table subscription" on page 34
- "Creating a DataPropagator Version 8 table subscription" on page 32

Related reference:

- "Data source restrictions for DataPropagator subscriptions" on page 39

Batching synchronization requests

Procedure:

You can use the XML Scripting tool to batch synchronization request. For more information on the XML Scripting tool see, "Utilizing the XML Scripting tool" on page 131.

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.

Related tasks:

- "Configuring DataPropagator for DB2 Everyplace subscriptions" on page 33
- "Associating the DataPropagator Subscription Set with a DB2 Everyplace table subscription" on page 34
- "Creating a DataPropagator Version 8 table subscription" on page 32

Related reference:

- "Data source restrictions for DataPropagator subscriptions" on page 39

Replicating on demand

The replication-on-demand feature enables you to request replication at any time. It is an alternative to setting replication to automatically run at regular intervals.

When the Sync Server servlet starts, the initial replication takes place as usual. Subsequent replications can be requested by using the `dsyreplicate` command.

Restrictions:

The following restrictions apply:

- If the `dsyreplicate` command must be issued from a machine different than the one that the Sync Server servlet runs on, both machines need to be configured to use the same Sync Server control database (DSYCTLDB).
- The `dsyreplicate` command must be issued for each mirror database.

- If a mirror database uses a DataPropogator subscription, the command must be issued on the same machine where the mirror database resides.

Procedure:

1. Open a DB2 UDB command window.
2. On Windows, change to %DSYINSTDIR%\Server\bin. On UNIX, change to $DSYINSTDIR/Server/bin.
3. Issue the `dsyreplicate`command to request the replication..

   ```
   dsyreplicate [-timeout timeout] mirror_database_name
   ```

   where *timeout* specifies the maximum amount of time (in seconds) that `dsyreplicate` will wait for the Sync Servers to quiesce all synchronization activities. It does *not* specify the time limit for replication. There is no time limit for replication. That is, the more data you have, the longer replication will take.

   For example, dsyreplicate m_vnurse

   Note: The default timeout is 5 seconds.
4. Close the DB2 command window.

Related tasks:

- "Configuring DataPropagator for DB2 Everyplace subscriptions" on page 33
- "Associating the DataPropagator Subscription Set with a DB2 Everyplace table subscription" on page 34
- "Creating a DataPropagator Version 8 table subscription" on page 32

Related reference:

- "Data source restrictions for DataPropagator subscriptions" on page 39

Filtering data from data sources

This chapter provides information about how to filter data from data sources using the Sync Server. The topics covered are:

- "Filtering data from data sources"
- "Filtering data horizontally" on page 42
- "Filtering data vertically" on page 47

Filtering data from data sources

DB2 Everyplace Sync Server includes several filtering options for `horizontal–`, `vertical–`, and `user–based` filtering methods. Filtering data from the server is an important part of controlling the data available to the client. Filtering can also help control client database size by synchronizing the minimum amount of data necessary for a specific client. The following topics show you how to use the DB2 Everyplace Sync Server filtering methods for:

- Users
- Groups

Related tasks:

- "Filtering data at the subscription level" on page 43
- "Filtering data horizontally" on page 42
- "Defining simple horizontal filters" on page 43

- “Filtering data at the group level” on page 44
- “Filtering data at the user level” on page 45
- “Defining complex horizontal filters” on page 46
- “Filtering data vertically” on page 47

Related reference:

- “Syntax for filters” on page 202

Filtering data horizontally

This chapter provides information about how to horizontally filter data from data sources using the Sync Server. The topics covered are:

- “Filtering data horizontally”
- “Defining simple horizontal filters” on page 43
- “Filtering data at the subscription level” on page 43
- “Filtering data at the group level” on page 44
- “Filtering data at the user level” on page 45
- “Defining complex horizontal filters” on page 46

Filtering data horizontally

Horizontal filtering at the subscription level, group level, or user level uses an SQL clause to select data from the source data source. The contents of the SQL clause are controlled by the Rows page of the **Advanced Subscription Definition** notebook for DB2 DataPropagator subscriptions and the Rows page of **Advanced Replication Subscription** notebook for JDBC subscriptions. To open either of these notebooks:

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 notebook opens. For JDBC subscriptions, the Advanced Replication Subscription notebook 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.

This topic show you how to define two types of horizontal filters:

- “Defining simple horizontal filters” on page 43
- “Defining complex horizontal filters” on page 46

Related tasks:

- “Filtering data at the subscription level” on page 43
- “Filtering data from data sources” on page 41
- “Defining simple horizontal filters” on page 43
- “Filtering data at the group level” on page 44
- “Filtering data at the user level” on page 45
- “Defining complex horizontal filters” on page 46
- “Filtering data vertically” on page 47

Related reference:

- “Syntax for filters” on page 202

Defining simple horizontal filters

You can define several types of horizontal filters:

- “Filtering data at the subscription level”
- “Filtering data at the group level” on page 44
- “Filtering data at the user level” on page 45

Related tasks:

- “Filtering data at the subscription level”
- “Filtering data from data sources” on page 41
- “Filtering data horizontally” on page 42
- “Filtering data at the group level” on page 44
- “Filtering data at the user level” on page 45
- “Defining complex horizontal filters” on page 46
- “Filtering data vertically” on page 47

Related reference:

- “Syntax for filters” on page 202

Filtering data at the subscription level

Procedure:

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 notebook opens. For JDBC subscriptions, the Advanced Replication Subscription notebook 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.

On the rows page of the subscription notebook, 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'
```

Related tasks:

- “Filtering data from data sources” on page 41
- “Filtering data horizontally” on page 42
- “Defining simple horizontal filters”
- “Filtering data at the group level” on page 44
- “Filtering data at the user level” on page 45
- “Defining complex horizontal filters” on page 46
- “Filtering data vertically” on page 47
- “Filtering data vertically” on page 47

Related reference:

- "Syntax for filters" on page 202

Filtering data at the group level

Horizontal filtering at the subscription level, group level, or user level uses an SQL clause to select data from the source data source. The contents of the SQL clause are controlled by the Rows page of the **Advanced Subscription Definition** notebook for DB2 DataPropagator subscriptions and the Rows page of Advanced Replication Subscription notebook for JDBC subscriptions.

Procedure:

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 notebook opens. For JDBC subscriptions, the Advanced Replication Subscription notebook opens.
4. On the Rows page, type an SQL clause in the **Subset of rows for individual users** box. Do not type WHERE at the beginning of the SQL clause. WHERE is automatically appended to the clause typed in the box. Instead of using a value in the SQL clause, insert a parameter. For example, to filter the VNPERSON table based on the value of the parameter **:fcity** for the city column and the value of the parameter **:fid** for the id column, type:

   ```
   city=':fcity' and id=':fid'
   ```

   where **:fcity** is a parameter for the city column value and **:fid** is a parameter for the id column value.
5. Complete and close the Create Subscription notebook.
6. Open a Create or Edit Group notebook for the group of users you want to filter. The subscription you created with the filter parameters must be assigned to the group you are editing.
7. On the Data filter page of the Group notebook, click **Add**.
8. Type `:fcity` in the **Parameter name** field.
9. Type a default value in the **Default value** field. For example:

   ```
   Los Angeles, CA 90061
   ```
10. Click **OK**.
11. Click **Add** again to add the second parameter.
12. Type `:fid.` in the **Parameter name** field.
13. Click **OK**.
14. Complete and close the Group notebook.
15. Open a Create or Edit User notebook for the user you want to filter. The user must be assigned to the group with the filter parameters.
16. On the Data filter page of the Group notebook, select the **:fid** parameter and click **Change**.
17. Type a value in the **User override** field. For example, to override the value of **:fid** for this user and set it to `900000401`, type:

   ```
   900000401
   ```
18. Click **OK**.

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.

Related tasks:

- "Filtering data at the subscription level" on page 43
- "Filtering data from data sources" on page 41
- "Filtering data horizontally" on page 42
- "Defining simple horizontal filters" on page 43
- "Filtering data at the user level"
- "Defining complex horizontal filters" on page 46
- "Filtering data vertically" on page 47

Related reference:

- "Syntax for filters" on page 202

Filtering data at the user level

Horizontal filtering at the subsription level, group level, or user level uses an SQL clause to select data from the source data source. The contents of the SQL clause are controlled by the Rows page of the Advanced Subscription Definition notebook for DB2 DataPropagator subscriptions and the Rows page of Advanced Replication Subscription notebook for JDBC subscriptions.

Procedure:

1. On the Rows page, type an SQL clause in the **Subset of rows for individual users** box. Do not type WHERE at the beginning of the SQL clause. WHERE is automatically appended to the clause typed in the box. Instead of using a value in the SQL clause, insert a parameter. For example, to filter the VNPERSON table based on the value of the parameter **:fcity** for the city column and the value of the parameter **:fid** for the id column, type:

   ```
   city=':fcity' and id=':fid'
   ```

   where **:fcity** is a parameter for the city column value and **:fid** is a parameter for the id column value.
2. Complete and close the Create Subscription notebook.
3. Open a Create or Edit Group notebook for the group of users you want to filter. The subscription you created with the filter parameters must be assigned to the group you are editing.
4. On the Data filter page of the Group notebook, click **Add**.
5. Type `:fcity` in the **Parameter name** field.
6. Type a default value in the **Default value** field. For example:

   ```
   Los Angeles, CA 90061
   ```
7. Click **OK**.
8. Click **Add** again to add the second parameter.
9. Type `:fid.` in the **Parameter name** field.
10. Click **OK**.
11. Complete and close the Group notebook.
12. Open a Create or Edit User notebook for the user you want to filter. The user must be assigned to the group with the filter parameters.
13. On the Data filter page of the Group notebook, select the **:fid** parameter and click **Change**.
14. Type a value in the **User override** field. For example, to override the value of **:fid** for this user and set it to `900000401`, type:

    ```
    900000401
    ```

15. Click **OK**.

The user will only receive records with a city column value of `Los Angeles, CA 90061` and an id column value of `900000401`. There is one record in VNPERSON with this value. Other users assigned to the same group will only receive rows from the VNPERSON table with a city column value equal to `Los Angeles, CA 90061` and the user column value set by their data filters. Other groups will receive the entire VNPERSON table or a specific subset of the VNPERSON table, depending on the filters configured.

Related tasks:

- "Filtering data at the subscription level" on page 43
- "Filtering data from data sources" on page 41
- "Filtering data horizontally" on page 42
- "Defining simple horizontal filters" on page 43
- "Filtering data at the group level" on page 44
- "Defining complex horizontal filters"
- "Filtering data vertically" on page 47

Related reference:

- "Syntax for filters" on page 202

Defining complex horizontal filters

Some situations in database and user management call for the definition of a complex SQL WHERE clause in **Subset of rows for individual users** field in order to limit the rows that a group or a user can see. When defining the filter, you might refer to a table in the mirror database, use parameters for the group or the user, or use a combination of these methods. For the basic steps in creating a simple horizontal filter, see "Filtering data at the subscription level" on page 43.

DB2 Everyplace Sync Server provides a subset of the standard SQL WHERE implementation to use with the filter. For the syntax of the subset, see "Syntax for filters" on page 202.

The following example demonstrates how you may refer to other tables in the mirror database and use a parameter for horizontal filtering.

Suppose you manage two tables named Ziptab and Business, both of which are in the mirror database. Ziptab contains ZIP codes for certain cities (see Table 2), while Business keeps track of some companies and their ZIP codes (see Table 3). You want users in San Jose, California to view only the businesses in their city. You have created a group and assigned these users to the group.

Table 2. Ziptab table

City	Zipcode
San Jose	95141
San Jose	95123
Los Angeles	93002

Table 3. Business table

Business	Zip
IBM	95141

Table 3. Business table (continued)

Business	Zip
My Company	95123
Your Company	93002
Another Company	94888

You would type the following clause in the **Subset of rows for individual users** field of the Rows page:

```
(Zip) IN (SELECT Zipcode FROM Ziptab WHERE City = ':fcity')
```

:fcity is a parameter for the City column value, which you set to `San Jose` for the group.

Then complete and close the Create Subscriptions notebook. As a result, after synchronization, the users will only see the following rows in Business table on their devices:

Table 4. Rows that the users will see on their devices

Business	Zip
IBM	95141
My Company	95123

Related tasks:

- "Filtering data at the subscription level" on page 43
- "Filtering data from data sources" on page 41
- "Filtering data horizontally" on page 42
- "Defining simple horizontal filters" on page 43
- "Filtering data at the group level" on page 44
- "Filtering data at the user level" on page 45
- "Filtering data vertically"

Related reference:

- "Syntax for filters" on page 202

Filtering data vertically

Vertical filtering uses the Target columns page of the Advanced Subscription Definition notebook for DB2 DataPropagator subscriptions and the Rows page of Advanced Replication Subscription notebook for JDBC subscription. To open either of these notebooks:

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 notebook opens. For JDBC subscriptions, the Advanced Replication Subscription notebook opens.

This section uses a sample table with 7 columns. The characteristics of this sample table are:

```
Column name    data type  nullable?    default value
NOTNULLINT1    INTEGER    NO
NOTNULLCHAR1   CHAR       NO
NOTNULLDATE1   DATE       NO
DEFAULTINT1    INT        YES          1000
DEFAULTCHAR1   CHAR       YES          'a'
DEFAULTDATE1   DATE       YES          '2001-06-29'
ID             INTEGER    NO
```

The ID column is the primary key.

Procedure:

To vertically filter data:

1. On the Target Columns page, unsubscribe the columns you do not want to synchronize to the client. To unsubscribe a column, clear the **Subscribe** check box. For example, using the sample table, clear the **Subscribe** check box for the first four columns: `NOTNULLINT1`, `NOTNULLCHAR1`, `NOTNULLDATE1`, and `DEFAULTINT1`.
2. Select each unsubscribed column, and click **Change** to add a default value data filter. The Change Column window opens.
3. Type a default value data filter parameter in the **Default value data filter** field. Repeat for all unsubscribed columns. The example table uses the parameters: `:PARAMINT1`, `:PARAMCHAR1`, `:PARAMDATE1`, and `:PARAMINT2`.
4. Close the Subscription notebook.

The default value data–filter parameters are used by the group to provide a default value for the unsubscribed columns. Data inserted to the data source from the client will be composed of the columns of data on the client combined with default column values defined for the group to complete the rows of data inserted to the data source. The next step is to define the default value data filter parameter values.

Related tasks:

- "Filtering data at the subscription level" on page 43
- "Filtering data from data sources" on page 41
- "Filtering data horizontally" on page 42
- "Defining simple horizontal filters" on page 43
- "Filtering data at the group level" on page 44
- "Filtering data at the user level" on page 45
- "Defining complex horizontal filters" on page 46

Related reference:

- "Syntax for filters" on page 202

Creating an application using the remote query and stored procedure adaptor

This chapter provides information about how to create an application using the remote query and stored procedure adaptor. The topics covered are:

- "Creating an application using the remote query and stored procedure adapter" on page 49
- "The remote query and stored procedure adapter" on page 50
- "Supported data types for stored procedures" on page 50

- “CLI functions and restrictions for result sets” on page 51
- “Using the remote query and stored procedure adaptor sample application” on page 51
- “Sample application code” on page 56

Creating an application using the remote query and stored procedure adapter

The following example show you how to create a stored procedure, a subscription to the stored procedure, and a DB2 Everyplace application to use the stored procedure. The purpose of the example application is to allow a mobile user to check an account balance and transfer money between a savings and a checking account by using a DB2 Everyplace remote stored procedure call.

Prerequisites:

The remote query and stored procedure adapter supports the Win32 (non-unicode), WinCE, Symbian, and Palm OS client platforms. The remote query and stored procedure adapter requires that the stored procedures are registered to DB2.

Restrictions:

- Multiple connections

  DB2 Everyplace supports one connection to the local database and one connection to the remote database with some limitations. The application must connect to the local database first (if a local connection is needed), then connect to the remote database (if a remote connection is needed). The allocated connection handles should be freed together.
- Statement handle

  User should allocate only one statement handle for the remote connection.
- On Palm OS, the application stack size may need to be increased.
- On Win32 operating systems, at run time the IBM Sync Client DLL files must be included in the local directory or system path.
- In a DB2 Universal Database stored procedure, if a binary large object (BLOB) is used as an input or output parameter, the first four bytes of the BLOB data are reserved to indicate length.

Procedure:

To create a stored procedure, a subscription to the stored procedure, and a DB2 Everyplace application to use the stored procedure using the sample application:

1. “Creating a stored procedure using the sample application” on page 51
2. “Creating a DB2 Everyplace application” on page 55

Related concepts:

- “The remote query and stored procedure adapter” on page 50

Related tasks:

- “Creating the Agent Adapter subscription for the sample application” on page 54

Related reference:

- “Sample application code” on page 56

- “Supported data types for stored procedures”
- “CLI functions and restrictions for result sets” on page 51

The remote query and stored procedure adapter

DB2 Everyplace includes a remote query and stored procedure adapter. This adapter enables DB2 Everyplace application to use the Sync Server architecture to call a stored procedure located at a remote data source. The results of the stored procedure are returned directly to the application on the device. The stored procedure call allows a DB2 Everyplace application to directly access data in a remote server without synchronizing.

The remote query and stored procedure adapter enables some unique capabilities of the DB2 Everyplace database engine. This topic details the requirements and techniques for using the remote query and stored procedure adapter in a DB2 Everyplace application. These include:

- “Supported data types for stored procedures”
- “CLI functions and restrictions for result sets” on page 51

Related tasks:

- “Creating the Agent Adapter subscription for the sample application” on page 54
- “Creating an application using the remote query and stored procedure adapter” on page 49

Related reference:

- “Sample application code” on page 56
- “Creating a stored procedure using the sample application” on page 51
- “Creating a DB2 Everyplace application” on page 55
- “Supported data types for stored procedures”
- “CLI functions and restrictions for result sets” on page 51

Supported data types for stored procedures

DB2 Everyplace supports calling stored procedures on a remote DB2 server through the CLI or JDBC interface. The client application uses the CALL statement to run the remote stored procedure. The CALL statement names the procedure to be called and specifies its parameters. The following types are supported: INTEGER, SMALLINT, DECIMAL, CHAR, VARCHAR, DATE, TIME, TIMESTAMP and BLOB.

Related concepts:

- “The remote query and stored procedure adapter”

Related tasks:

- “Creating the Agent Adapter subscription for the sample application” on page 54
- “Creating an application using the remote query and stored procedure adapter” on page 49

Related reference:

- “Sample application code” on page 56
- “Creating a stored procedure using the sample application” on page 51

- “Creating a DB2 Everyplace application” on page 55
- “CLI functions and restrictions for result sets”

CLI functions and restrictions for result sets

Result sets are a useful way to retrieve data from a stored procedure. If a client application runs a stored procedure that generates a result set, it can then use the regular CLI functions or JDBC methods such as `SQLFetch()` and `SQLGetData()` to retrieve the data. DB2 Everyplace does not support multiple result sets.

Related concepts:

- “The remote query and stored procedure adapter” on page 50

Related tasks:

- “Creating the Agent Adapter subscription for the sample application” on page 54
- “Creating an application using the remote query and stored procedure adapter” on page 49

Related reference:

- “Sample application code” on page 56
- “Creating a stored procedure using the sample application”
- “Creating a DB2 Everyplace application” on page 55
- “Supported data types for stored procedures” on page 50

Using the remote query and stored procedure adaptor sample application

This chapter provides information about how to use the remote query and stored procedure adaptor sample application. The topics covered are:

- “Create a data source”
- “Creating a stored procedure using the sample application”
- “Creating the Agent Adapter subscription for the sample application” on page 54
- “Creating a DB2 Everyplace application” on page 55

Create a data source

This example uses a DB2 database named `MYSAMPLE`. You need to manually create the `MYSAMPLE` database. To create the `MYSAMPLE` database, enter the following statements at the DB2 command prompt:

```
CREATE table db2e.MYACCOUNT ( Name char(16), Saving int, Checking int)
INSERT into db2e.MYACCOUNT values('Michael', 5000, 5000)
INSERT into db2e.MYACCOUNT values('Frank', 5000, 5000)
```

After creating the database, create a stored procedure to modify the data in the database.

Creating a stored procedure using the sample application

This example uses a stored procedure named `MYPROC()`. This procedure takes five parameters: Account Name, Option, Transfer Amount, Saving Balance, Checking Balance. The following list identifies the purpose of each of the parameters:

```
Account Name:  Input Parameter to identify the account.
      Option:  Input Parameter to determine what to do.
               There are three options:
               1: Check balance.
```

```
                    2: Transfer from saving to checking.
                    3: Transfer from checking to saving.
  Transfer Amount:  Input Parameter of the amount to transfer between checking
                    and saving.
   Saving Balance:  Output Parameter returning the balance of saving account
 Checking Balance:  Output Parameter returning the balance of checking account
```

The following code builds the stored procedure:

```
SQL_API_RC SQL_API_FN
myProc(char * szName, int * nCmd, int * nAmount, int * nSaving, int * nChecking)
{
 SQLHENV  henv;
 SQLHDBC  hdbc;
 SQLHSTMT hstmt;
 SQLRETURN rc;
 int  nRetSize;

 SQLCHAR  str1[]="select saving, checking from db2e.myaccount where name = ?";
 SQLCHAR  str2[]="update db2e.myaccount set saving=saving - ?,
          checking=checking + ? where name=?";
 SQLCHAR  str3[]="update db2e.myaccount set saving=saving + ?,
          checking=checking - ? where name=?";

 //****************************************************************
 //* Prepare connection and statement
 //****************************************************************
 rc = SQLAllocHandle( SQL_HANDLE_ENV, SQL_NULL_HANDLE, &henv);
 //checkerror
 rc = SQLAllocHandle( SQL_HANDLE_DBC, henv, &hdbc);
 //checkerror
 rc = SQLSetConnectAttr(hdbc, SQL_ATTR_AUTOCOMMIT, SQL_AUTOCOMMIT_OFF, SQL_NTS);
 //checkerror
     rc = SQLConnect(hdbc, NULL, SQL_NTS, NULL, SQL_NTS, NULL, SQL_NTS);
 //checkerror
 rc = SQLAllocHandle( SQL_HANDLE_STMT, hdbc, &hstmt);
 //checkerror

 //****************************************************************
 //* Update account
 //****************************************************************
 if ( *nCmd == 2 || *nCmd == 3 ){
  if ( *nCmd == 2 ){ //Transfer from saving to checking
   rc = SQLPrepare(hstmt, str2, SQL_NTS); //checkerror
  }
  if ( *nCmd == 3 ){ //Transfer from checking to saving
   rc = SQLPrepare(hstmt, str3, SQL_NTS); //checkerror
  }
  rc = SQLBindParameter(hstmt,
    1,
    SQL_PARAM_INPUT,
    SQL_C_LONG,
    SQL_INTEGER,
    0,
    0,
    (SQLPOINTER)nAmount,
    0,
    NULL ); //checkerror
  rc = SQLBindParameter(hstmt,
    2,
    SQL_PARAM_INPUT,
    SQL_C_LONG,
    SQL_INTEGER,
    0,
    0,
    (SQLPOINTER)nAmount,
    0,
    NULL ); //checkerror

  rc = SQLBindParameter(hstmt,
    3,
    SQL_PARAM_INPUT,
    SQL_C_CHAR,
    SQL_CHAR,
    0,
    0,
    (SQLPOINTER)szName,
    0,
    NULL ); //checkerror
  rc = SQLExecute(hstmt); //checkerror
 }

 //****************************************************************
 //* Retrieve account balance
 //****************************************************************
 rc = SQLPrepare(hstmt, str1, SQL_NTS); //checkerror
 rc = SQLBindParameter(hstmt,
   1,
   SQL_PARAM_INPUT,
   SQL_C_CHAR,
   SQL_CHAR,
   0,
   0,
   (SQLPOINTER)szName,
   0,
   NULL );//checkerror
 rc = SQLExecute(hstmt);//checkerror
 if ( rc == SQL_SUCCESS || rc == SQL_SUCCESS_WITH_INFO )
 {
  while ( (rc = SQLFetch(hstmt) ) == SQL_SUCCESS ){
   rc = SQLGetData( hstmt,
    (SQLSMALLINT)1,
    SQL_C_LONG,
    nSaving,
    sizeof(int) ,
    &nRetSize ) ; //checkerror
   rc = SQLGetData( hstmt,
    (SQLSMALLINT)2,
    SQL_C_LONG,
    nChecking,
    sizeof(int) ,
    &nRetSize ) ; //checkerror
  }
 }
 //****************************************************************
 //* Clean up
 //****************************************************************
     rc = SQLEndTran( SQL_HANDLE_DBC, hdbc, SQL_COMMIT );
 SQLFreeHandle(SQL_HANDLE_STMT, hstmt);
 SQLDisconnect(hdbc);
 SQLFreeHandle(SQL_HANDLE_DBC, hdbc);
 SQLFreeHandle(SQL_HANDLE_ENV, henv);
 return (0);
}
```

On the Win32 platform, after building the stored procedure into a dynamic link library (mydll.dll), copy it to the \SQLLIB\function directory. Next, register the stored procedure.

1. Open a DB2 command window.
2. Connect to the MYSAMPLE database using the following command:

   ```
   DB2 CONNECT TO MYSAMPLE
   ```

3. Register the stored procedure using a script named regscript.scr to configure options. The following code is used for this script:

```
CREATE PROCEDURE db2e.MYPROC (IN szName CHAR(16),
                              IN nCmd INTEGER,
                              IN nAmount INTEGER,
                              OUT nSaving INTEGER,
                              OUT nChecking INTEGER )
DYNAMIC RESULT SETS 1
LANGUAGE C
PARAMETER STYLE GENERAL
NO DBINFO
FENCED
MODIFIES SQL DATA
PROGRAM TYPE SUB
EXTERNAL NAME 'mydll!myProc'@
```

   To run the script, enter the following command:

```
db2 -td@ -vf regscript.scr
```

The stored procedure db2e.MYPROC is now configured. Next, create a subscription using the Mobile Devices Administration Center.

Related concepts:

- “The remote query and stored procedure adapter” on page 50

Related tasks:

- “Creating the Agent Adapter subscription for the sample application”
- “Creating an application using the remote query and stored procedure adapter” on page 49

Related reference:

- “Sample application code” on page 56
- “Creating a DB2 Everyplace application” on page 55
- “Supported data types for stored procedures” on page 50
- “CLI functions and restrictions for result sets” on page 51

Creating the Agent Adapter subscription for the sample application

Procedure:

1. Open the Administrator from the start menu.
2. Select the **Subscriptions** folder of the Administrator.
3. Right-click on the **Subscriptions** folder of the Administrator and select **Create Custom Subscription**.
4. Enter a name for the subscription in the **Name** field. Type `subex` in the **Name** field.
5. Select the AgentAdapter in the **Adapter** field.
6. Select None in the **Encryption** field. Encryption is not support with the AgentAdapter.
7. Click the **Launch Customizer** button. The Source database window opens.
8. Type a DB2 user ID with access privileges to the database into the **User ID** field
9. Type the password for the user ID in the **Password** and **Verify password** field.

10. In the **Other** field, type the following line:

    ```
    dbname=mysample;procname=db2e.MYPROC
    ```

    The **dbname** is the database used by the stored procedure. The **procname** is the name of the stored procedure.
11. Click **OK** to close the Source database window. Click **OK** to close the Create Custom Subscription notebook.

After creating the AgentAdapter subscription, create a user, group, and subscription set.

Related concepts:

- "The remote query and stored procedure adapter" on page 50

Related tasks:

- "Creating an application using the remote query and stored procedure adapter" on page 49

Related reference:

- "Sample application code" on page 56
- "Creating a stored procedure using the sample application" on page 51
- "Creating a DB2 Everyplace application"
- "Supported data types for stored procedures" on page 50
- "CLI functions and restrictions for result sets" on page 51

Creating a DB2 Everyplace application

This sample uses a DB2 Everyplace Win32 console application to test the remote query and stored procedure adapter. The sample application is called myclient.exe. The sample application uses the following three parameters:

```
    Account Name:   Identify the account to access.
          Option:   Identify the action to perform. The options are:
                       1: Check balance.
                       2: Transfer from savings to checking.
                       3: Transfer from checking to savings.
          Amount:   Amount to transfer between checking and saving.
```

For example, to transfer $1000 from savings to checking on the Michael account, enter the following command:

```
myclient.exe Michael 2 1000
```

The following response is returned:

```
 Saving = 4000
 Checking = 6000
```

Related concepts:

- "The remote query and stored procedure adapter" on page 50

Related tasks:

- "Creating the Agent Adapter subscription for the sample application" on page 54
- "Creating an application using the remote query and stored procedure adapter" on page 49

Related reference:

- “Sample application code”
- “Creating a stored procedure using the sample application” on page 51
- “Supported data types for stored procedures” on page 50
- “CLI functions and restrictions for result sets” on page 51

Sample application code

The following section contains the code in Java and C for the sample application. The code requires a connection string for the SQLConnect() function to connect to the remote data source.

The format of the connection string is:

```
http://IPAddr:port/db2e/servlet/com.ibm.mobileservices.adapter.agent.AgentServlet?DB=mysample
```

or

```
as http://IPAddr:port/db2e/agent?DB=mysample
```

where *IPAddr:port* is the IP address and port number of the server. For example:

```
http://192.168.0.11:8080/db2e/servlet/
com.ibm.mobileservices.adapter.agent.AgentServlet?DB=mysample
```

```
int main(int argc, char * argv[])
{
 SQLHENV  henv;
 SQLHDBC  hdbc;
 SQLHSTMT hstmt;
 SQLRETURN rc;
 SQLCHAR  strSQL[] = "CALL db2e.MYPROC(?,?,?,?,?)";
 int   nInd4, nInd5;
 int   nSaving = 0, nChecking =0 ;
 int   nCmd =0, nAmount=0;
 SQLCHAR  strConnect[254];

 //****************************************************************
 //* Check input parameters
 //****************************************************************
 if ( argc < 4 ){
  printf("\nUsage : myClient AccountName Cmd Amount");
  printf("\n     cmd 1 : query balance");
  printf("\n     cmd 2 : Transfer from Saving to Checking");
  printf("\n     cmd 3 : Trnasfer from Checking to Saving");
  return (99);
 }
 nCmd = atoi(argv[2]);
 nAmount = atoi(argv[3]);

 //****************************************************************
 //* Allocate handles
 //****************************************************************
 rc = SQLAllocHandle( SQL_HANDLE_ENV,
      SQL_NULL_HANDLE,
      &henv); //checkerror
 rc = SQLAllocHandle( SQL_HANDLE_DBC,
      henv,
      &hdbc); //checkerror
 if (argc == 5){
  strcpy(strConnect,"http://");
  strcat(strConnect,argv[4]);
  strcat(strConnect,"/db2e/servlet/
            com.ibm.mobileservices.adapter.agent.AgentServlet?DB=mysample");
 }else{
  strcpy(strConnect,"http://127.0.0.1:8080/db2e/servlet/
             com.ibm.mobileservices.adapter.agent.AgentServlet?DB=mysample");
 }

 //****************************************************************
 //* Connect to remote database
 //****************************************************************
 rc = SQLConnect(hdbc,
  strConnect,
  SQL_NTS,
  "userex", SQL_NTS,
  "userex", SQL_NTS  ); //checkerror
 rc = SQLAllocHandle( SQL_HANDLE_STMT,
     hdbc,
     &hstmt); //checkerror
 //*****************************************************************
 //* Prepare, Bind , and Execute the statement
 //*****************************************************************
 rc = SQLPrepare(hstmt,strSQL, SQL_NTS); //checkerror
 rc = SQLBindParameter(hstmt,
   1,
   SQL_PARAM_INPUT,
   SQL_C_CHAR,
   SQL_CHAR,
   0,
   0,
   (SQLPOINTER)argv[1],
   0,
   NULL ); //checkerror
 rc = SQLBindParameter(hstmt,
   2,
   SQL_PARAM_INPUT,
   SQL_C_LONG,
   SQL_INTEGER,
   0,
   0,
   (SQLPOINTER)&nCmd,
   sizeof(int),
   NULL); //checkerror
 rc = SQLBindParameter(hstmt,
   3,
   SQL_PARAM_INPUT,
   SQL_C_LONG,
   SQL_INTEGER,
   0,
   0,
   (SQLPOINTER)&nAmount,
   sizeof(int),
   NULL ); //checkerror
 rc = SQLBindParameter(hstmt,
   4,
   SQL_PARAM_OUTPUT,
   SQL_C_LONG,
   SQL_INTEGER,
   0,
   0,
   (SQLPOINTER)&nSaving,
   sizeof(int),
   &nInd4); //checkerror
 rc = SQLBindParameter(hstmt,
   5,
   SQL_PARAM_OUTPUT,
   SQL_C_LONG,
   SQL_INTEGER,
   0,
   0,
   (SQLPOINTER)&nChecking,
   sizeof(int),
   &nInd5 ); //checkerror
 rc = SQLExecute(hstmt); //checkerror
 //*****************************************************************
 //* Print the balance
 //*****************************************************************
 printf("\nSaving = %d",nSaving);
 printf("\nChecking = %d",nChecking);

 SQLFreeHandle(SQL_HANDLE_STMT, hstmt);
 SQLDisconnect(hdbc);
 SQLFreeHandle(SQL_HANDLE_DBC, hdbc);
 SQLFreeHandle(SQL_HANDLE_ENV, henv);
 return 0;
}
```

Sample application code in Java:

The following Java code has the same function as the C sample.

```
import java.sql.*;

class MyClient{
   public static void main(String [] args){

   if (args.length != 3){
            System.out.println("Usage java MyClient AccountName Cmd Amount");
            System.out.println("  cmd 1 : Query balance.");
            System.out.println("  cmd 2 : Transfer from Saving to Checking.");
            System.out.println("  cmd 3 : Transfer from Checking to Saving.");
            System.exit(-1);
        }

        String driver = "com.ibm.db2e.jdbc.DB2eDriver";
        String url = "jdbc:db2e:http://9.30.40.21:8080/db2e/servlet/
             com.ibm.mobileservices.adapter.agent.AgentServlet?DB=mysample";
        String sql = "Call db2e.MYPROC(?,?,?,?,?)";

        try{
            Class.forName(driver);
            //****************************************************************
            //* Connect to remote database
            //****************************************************************
            Connection con = DriverManager.getConnection(url,"userex","userex");

            CallableStatement cst =  con.prepareCall(sql);

            //****************************************************************
            //* Register the output parameters
            //****************************************************************
            cst.registerOutParameter (4, Types.INTEGER);
            cst.registerOutParameter (5, Types.INTEGER);

            //****************************************************************
            //* Set input parameters
            //****************************************************************
            cst.setString (1, args[0]);
            cst.setString (2, args[1]);
            cst.setString (3, args[2]);

            //****************************************************************
            //* Call the Remote Stored Procedure
            //****************************************************************
            cst.execute ();

            //****************************************************************
            //* Retrieve output
            //****************************************************************
            System.out.println("\nSaving = " + cst.getInt(4));
            System.out.println("\nChecking = " + cst.getInt(5));

            cst.close();
            con.close();
        }catch (SQLException sqlEx){
            while(sqlEx != null){
                System.out.println("SQLERROR: \n" + sqlEx.getErrorCode() +
                        ", SQLState: " + sqlEx.getSQLState() +
                        ", Message: " + sqlEx.getMessage() +
                        ", Vendor: " + sqlEx.getErrorCode() );
                sqlEx = sqlEx.getNextException();
            }
        }catch (Exception ex){
            ex.printStackTrace();
        }
    }
}
```

Related concepts:

- "The remote query and stored procedure adapter" on page 50

Related tasks:

- "Creating the Agent Adapter subscription for the sample application" on page 54
- "Creating an application using the remote query and stored procedure adapter" on page 49

Related reference:

- "Creating a stored procedure using the sample application" on page 51
- "Creating a DB2 Everyplace application" on page 55
- "Supported data types for stored procedures" on page 50
- "CLI functions and restrictions for result sets" on page 51

Part 3. Managing users and data

Chapter 4. Mobile Devices Administration Center and synchronization objects

This chapter provides conceptual information about Mobile Devices Administration Center and synchronization objects. The topics covered are:

- "Mobile Devices Administration Center and synchronization objects"
- "What is the Mobile Devices Administration Center?" on page 64
- "Starting the Mobile Devices Administration Center" on page 64
- "The Mobile Devices Administration Center interface" on page 65
- "What is a synchronization object?" on page 67
- "Where synchronization objects are stored" on page 69
- "Creating synchronization objects" on page 69
- "Creating an object from the object tree" on page 70
- "Creating a new object when working with another" on page 71

Mobile Devices Administration Center and synchronization objects

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 Administrator, 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.

Related concepts:

- "What is the Mobile Devices Administration Center?" on page 64
- "The Mobile Devices Administration Center interface" on page 65
- "What is a synchronization object?" on page 67
- "Where synchronization objects are stored" on page 69
- "Information displayed in the contents pane" on page 71

Related tasks:

- "Starting the Mobile Devices Administration Center" on page 64
- "Creating an object from the object tree" on page 70
- "Creating synchronization objects" on page 69
- "Refreshing the view" on page 73
- "Filtering the contents pane" on page 73
- "Editing a synchronization object" on page 74
- "Deleting an object" on page 75
- "Creating a new object when working with another" on page 71

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 Administrator 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.

Related concepts:

- "Mobile Devices Administration Center and synchronization objects" on page 63
- "The Mobile Devices Administration Center interface" on page 65
- "What is a synchronization object?" on page 67
- "Where synchronization objects are stored" on page 69
- "Information displayed in the contents pane" on page 71

Related tasks:

- "Starting the Mobile Devices Administration Center"
- "Creating an object from the object tree" on page 70
- "Creating synchronization objects" on page 69
- "Refreshing the view" on page 73
- "Filtering the contents pane" on page 73
- "Editing a synchronization object" on page 74
- "Deleting an object" on page 75
- "Creating a new object when working with another" on page 71

Starting the Mobile Devices Administration Center

Prerequisites:

- For UNIX, you must be logged in as the DB2 Everyplace Sync Server instance in order to start the Mobile Devices Administration Center.
- For Linux, DB2 JDBC Applet Server needs to be started before starting the Mobile Devices Administration Center. Use the following command:

  ```
  db2jstrt
  ```

  Optional: You can specify a port for starting the DB2 JDBC Applet Server. The default port is 6789. Be sure to restart the DB2 JDBC Applet Server every time you restart the machine.

Procedure:

To open the Mobile Devices Administration Center and DB2 Control Center For Windows, click **Start —> Programs —> IBM DB2 Everyplace—> Start Mobile Devices Administration Center**.

Note: The DISPLAY environment variable has to be set in the shell in which the `dsyadmin.sh` command is issued. For example: `export DISPLAY=:0.0`

To open the Mobile Devices Administration Center and DB2 Control Center on UNIX, change the directory to `$DSYINSTDIR/Server/bin`, then run `dsyadmin.sh`.

Note: On Linux, if you have started the DB2 JDBC Applet Server on a port other than 6789, you must provide that port as a parameter to dsyadmin.sh.

Related concepts:

- "Mobile Devices Administration Center and synchronization objects" on page 63
- "What is the Mobile Devices Administration Center?" on page 64
- "The Mobile Devices Administration Center interface"
- "What is a synchronization object?" on page 67
- "Where synchronization objects are stored" on page 69
- "Information displayed in the contents pane" on page 71

Related tasks:

- "Creating an object from the object tree" on page 70
- "Creating synchronization objects" on page 69
- "Refreshing the view" on page 73
- "Filtering the contents pane" on page 73
- "Editing a synchronization object" on page 74
- "Deleting an object" on page 75
- "Creating a new object when working with another" on page 71

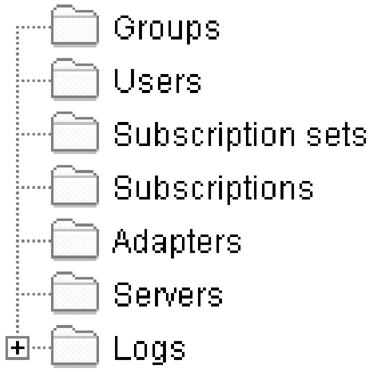
The Mobile Devices Administration Center interface

The Mobile Devices Administration Center is very similar to the DB2 Control Center. This section explains aspects of the interface that you will encounter in the main window and within object notebooks.

Controls in the main window:

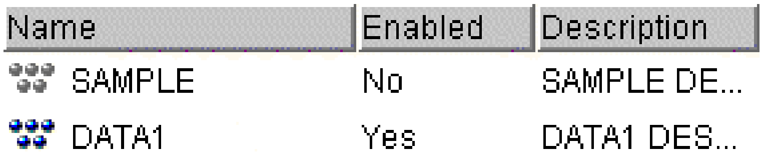
The main window contains an object tree and a contents pane to help you manage objects.

Object tree



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

Contents pane

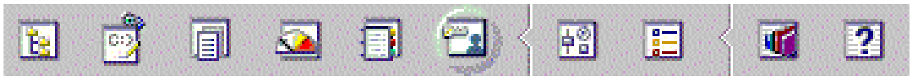


Name	Enabled	Description
SAMPLE	No	SAMPLE DE...
DATA1	Yes	DATA1 DES...

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.

Mobile Devices Administration Center toolbar

Note: The toolbar is not available in the Mobile Devices Administration Center if you are using DB2 Universal Database Version 8. Use the DB2 Universal Database Control Center to utilize the functions provided in the toolbar.

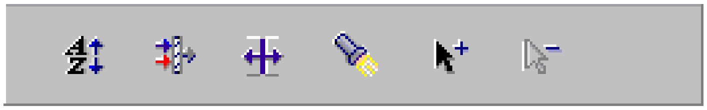


This toolbar is similar to the toolbar found in the DB2 Control Center. By clicking icons in this toolbar, you can work within different areas of the Control Center (such as the Command Center or Script Center), schedule and run jobs, change tools settings, and access help.

For more information about the Control Center toolbar, see the Control Center help.

On AIX, to view the help file for the DB2 Control Center, open `~/sqllib/doc/html/db2help/index.htm` in an HTML browser. To view the help file for the Administrator, open `~/sqllib/doc/html/db2help/dsysync/dsyhmsttfrm.htm` in an HTML browser.

Contents pane toolbar



Use the toolbar below the contents pane to tailor the view of synchronization objects and information in the contents pane to suit your needs.

Entry fields:

Required fields in notebooks and windows 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 custom controls with the keyboard:

You can use the keyboard to access controls found on the graphical user interface. The following control may be unique to DB2.

… To access this push button using the keyboard, use the Tab key until the push button is selected then press Enter.

Related concepts:

- “Mobile Devices Administration Center and synchronization objects” on page 63
- “What is the Mobile Devices Administration Center?” on page 64
- “What is a synchronization object?”
- “Where synchronization objects are stored” on page 69
- “Information displayed in the contents pane” on page 71

Related tasks:

- “Starting the Mobile Devices Administration Center” on page 64
- “Creating an object from the object tree” on page 70
- “Creating synchronization objects” on page 69
- “Refreshing the view” on page 73
- “Filtering the contents pane” on page 73
- “Editing a synchronization object” on page 74
- “Deleting an object” on page 75
- “Creating a new object when working with another” on page 71

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 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 an *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.

Tip: 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 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. Monitoring log activity is explained in "Viewing the error log to diagnose problems" on page 152.

Related concepts:

- "Mobile Devices Administration Center and synchronization objects" on page 63
- "What is the Mobile Devices Administration Center?" on page 64
- "The Mobile Devices Administration Center interface" on page 65
- "Where synchronization objects are stored" on page 69
- "Information displayed in the contents pane" on page 71

Related tasks:

- "Starting the Mobile Devices Administration Center" on page 64
- "Creating an object from the object tree" on page 70
- "Creating synchronization objects" on page 69
- "Refreshing the view" on page 73
- "Filtering the contents pane" on page 73
- "Editing a synchronization object" on page 74
- "Deleting an object" on page 75
- "Creating a new object when working with another" on page 71

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 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.

Related concepts:

- "Mobile Devices Administration Center and synchronization objects" on page 63
- "What is the Mobile Devices Administration Center?" on page 64
- "The Mobile Devices Administration Center interface" on page 65
- "What is a synchronization object?" on page 67
- "Information displayed in the contents pane" on page 71

Related tasks:

- "Starting the Mobile Devices Administration Center" on page 64
- "Creating an object from the object tree" on page 70
- "Creating synchronization objects"
- "Refreshing the view" on page 73
- "Filtering the contents pane" on page 73
- "Editing a synchronization object" on page 74
- "Deleting an object" on page 75
- "Creating a new object when working with another" on page 71

Creating synchronization objects

Procedure:

Although each object requires different input, you create all objects using the same basic process in the Mobile Devices Administration Center. You can create an object from the object tree or when you are working within another object.

- "Creating an object from the object tree" on page 70
- "Creating a new object when working with another" on page 71

Related concepts:

- "Mobile Devices Administration Center and synchronization objects" on page 63
- "What is the Mobile Devices Administration Center?" on page 64
- "The Mobile Devices Administration Center interface" on page 65
- "What is a synchronization object?" on page 67
- "Where synchronization objects are stored"

- “Information displayed in the contents pane” on page 71

Related tasks:

- “Starting the Mobile Devices Administration Center” on page 64
- “Refreshing the view” on page 73
- “Filtering the contents pane” on page 73
- “Editing a synchronization object” on page 74
- “Deleting an object” on page 75

Creating an object from the object tree

Procedure:

To create a synchronization object:

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.

The Create notebook opens with the object type displayed in the title of the notebook.

For details on how to complete the notebook pages for each object, see one of the following topics:

- For groups, see Creating a data synchronization group.
- For users, see Creating users in the Mobile Devices Administration Center.
- For subscriptions, see Creating a JDBC subscription.
- For subscriptions sets, seeDefining a subscription set.

After you save an object, you can edit the object to complete it or make changes. See Editing a synchronization object for more information.

Related concepts:

- “Mobile Devices Administration Center and synchronization objects” on page 63
- “What is the Mobile Devices Administration Center?” on page 64
- “The Mobile Devices Administration Center interface” on page 65
- “What is a synchronization object?” on page 67
- “Where synchronization objects are stored” on page 69
- “Information displayed in the contents pane” on page 71

Related tasks:

- “Starting the Mobile Devices Administration Center” on page 64
- “Creating synchronization objects” on page 69
- “Refreshing the view” on page 73
- “Filtering the contents pane” on page 73
- “Editing a synchronization object” on page 74
- “Deleting an object” on page 75
- “Creating a new object when working with another” on page 71

Creating a new object when working with another

Procedure:

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. Without leaving the Edit Group notebook, click **Create** on the Users page to open the Create User notebook, add the new nurse and then assign the nurse to the Visiting Nurses group.

Related concepts:

- "Mobile Devices Administration Center and synchronization objects" on page 63
- "What is the Mobile Devices Administration Center?" on page 64
- "The Mobile Devices Administration Center interface" on page 65
- "What is a synchronization object?" on page 67
- "Where synchronization objects are stored" on page 69
- "Information displayed in the contents pane"

Related tasks:

- "Starting the Mobile Devices Administration Center" on page 64
- "Creating synchronization objects" on page 69
- "Refreshing the view" on page 73
- "Filtering the contents pane" on page 73
- "Editing a synchronization object" on page 74
- "Deleting an object" on page 75

Mobile Devices Administration Center and contents pane

This section provides conceptual information about the Mobile Devices Administration Center contents pane. The topics covered are:

- "Information displayed in the contents pane"
- "Refreshing the view" on page 73
- "Filtering the contents pane" on page 73
- "Editing a synchronization object" on page 74
- "Deleting an object" on page 75

Information displayed in the contents pane

After you create an object, information about that object appears in the contents pane. 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 custom 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. You 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. (See "Enabling and disabling synchronization" on page 128 for information on enabling a group or user for synchronization.)

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.

Related concepts:

- "Mobile Devices Administration Center and synchronization objects" on page 63
- "What is the Mobile Devices Administration Center?" on page 64
- "The Mobile Devices Administration Center interface" on page 65
- "What is a synchronization object?" on page 67
- "Where synchronization objects are stored" on page 69

Related tasks:

- "Starting the Mobile Devices Administration Center" on page 64
- "Creating an object from the object tree" on page 70

- "Creating synchronization objects" on page 69
- "Refreshing the view"
- "Filtering the contents pane"
- "Editing a synchronization object" on page 74
- "Deleting an object" on page 75
- "Creating a new object when working with another" on page 71

Refreshing the view

When you make changes to an object, those changes might not be immediately displayed in the contents pane.

Procedure:

To refresh the contents of an object folder:

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

Related concepts:

- "Mobile Devices Administration Center and synchronization objects" on page 63
- "What is the Mobile Devices Administration Center?" on page 64
- "The Mobile Devices Administration Center interface" on page 65
- "What is a synchronization object?" on page 67
- "Where synchronization objects are stored" on page 69
- "Information displayed in the contents pane" on page 71

Related tasks:

- "Starting the Mobile Devices Administration Center" on page 64
- "Creating an object from the object tree" on page 70
- "Creating synchronization objects" on page 69
- "Filtering the contents pane"
- "Editing a synchronization object" on page 74
- "Deleting an object" on page 75
- "Creating a new object when working with another" on page 71

Filtering the contents pane

If you have several synchronization objects in a folder, you might need to 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.

Procedure:

To filter a folder's contents:

1. Right-click the object folder.
2. Select **Filter**. The Filter notebook 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 page, type your search criteria.

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.

For more information about object filters, see the online help for the DB2 Control Center.

Sorting the contents pane

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.

Related concepts:

- "Mobile Devices Administration Center and synchronization objects" on page 63
- "What is the Mobile Devices Administration Center?" on page 64
- "The Mobile Devices Administration Center interface" on page 65
- "What is a synchronization object?" on page 67
- "Where synchronization objects are stored" on page 69
- "Information displayed in the contents pane" on page 71

Related tasks:

- "Starting the Mobile Devices Administration Center" on page 64
- "Creating an object from the object tree" on page 70
- "Creating synchronization objects" on page 69
- "Refreshing the view" on page 73
- "Editing a synchronization object"
- "Deleting an object" on page 75
- "Creating a new object when working with another" on page 71

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.

Procedure:

To edit an object:

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 Edit notebook opens with the object type displayed in the title of the notebook. The Edit notebook contains all the same fields as the Create notebook for the object. For information on specific pages and fields, see the sections for the type of object that you want to edit:
   - To edit a group, see "Creating a data synchronization group" on page 79.
   - To edit a user, see "Creating users in the Mobile Devices Administration Center" on page 85.
   - To edit a subscription, see "Creating a data synchronization group" on page 79.

   Note: 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.

- To edit a subscription set, see "Enabling groups to access their data and files" on page 125.

Related concepts:

- "Mobile Devices Administration Center and synchronization objects" on page 63
- "What is the Mobile Devices Administration Center?" on page 64
- "The Mobile Devices Administration Center interface" on page 65
- "What is a synchronization object?" on page 67
- "Where synchronization objects are stored" on page 69
- "Information displayed in the contents pane" on page 71

Related tasks:

- "Starting the Mobile Devices Administration Center" on page 64
- "Creating an object from the object tree" on page 70
- "Creating synchronization objects" on page 69
- "Refreshing the view" on page 73
- "Filtering the contents pane" on page 73
- "Deleting an object"
- "Creating a new object when working with another" on page 71

Deleting an object

You can delete any object in the Mobile Devices Administration Center. 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. (See "Enabling and disabling synchronization" on page 128 for more information.)

Procedure:

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**.

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.

Related concepts:

- "Mobile Devices Administration Center and synchronization objects" on page 63
- "What is the Mobile Devices Administration Center?" on page 64
- "The Mobile Devices Administration Center interface" on page 65

- “What is a synchronization object?” on page 67
- “Where synchronization objects are stored” on page 69
- “Information displayed in the contents pane” on page 71

Related tasks:

- “Starting the Mobile Devices Administration Center” on page 64
- “Creating an object from the object tree” on page 70
- “Creating synchronization objects” on page 69
- “Refreshing the view” on page 73
- “Filtering the contents pane” on page 73
- “Editing a synchronization object” on page 74
- “Creating a new object when working with another” on page 71

Chapter 5. Using Mobile Devices Administration Center

This chapter provides information about how to use the Mobile Devices Administration Center. The topics covered are:

- "Using the Mobile Devices Administration Center"
- "Evalutating the characteristics of your mobile users" on page 78
- "Creating a data synchronization group" on page 79
- "Defining mobile users to the Sync Server" on page 85
- "Setting up data and replication sets" on page 90
- "Enabling groups to access their data" on page 125
- "Enabling and disabling synchronization" on page 128

Using the Mobile Devices Administration Center

The following topics explain how to use the Mobile Devices Administration Center to set up groups of users, and to configure data and file replication for groups. The topics include:

- Characteristics of your mobile users to determine how to separate them into groups.
- Creating a data synchronization group.
- Creating users in the Mobile Devices Administration Center.
- Set up data and file replication
  - Creating a JDBC subscription
  - Creating a DataPropagator subscription
  - Creating an upload subscription
  - Creating a file subscription
  - Creating a custom subscription
  - Creating a custom adapter
- Enabling groups to access their data and files.
- Enabling and disabling synchronization.

Related concepts:

- "Evalutating the characteristics of your mobile users" on page 78
- "Data set up and replication" on page 90

Related tasks:

- "Creating a data synchronization group" on page 79
- "Creating users in the Mobile Devices Administration Center" on page 85
- "Enabling groups to access their data and files" on page 125
- "Enabling and disabling synchronization" on page 128

Evalutating the characteristics of your mobile users

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 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 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.

Related concepts:

- "Data set up and replication" on page 90

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Creating a data synchronization group" on page 79
- "Creating users in the Mobile Devices Administration Center" on page 85
- "Enabling groups to access their data and files" on page 125
- "Enabling and disabling synchronization" on page 128

Creating a data synchronization group

This section provides information about how to create a data synchronization group using the Mobile Devices Administration Center. The topics covered are:

- “Creating a data synchronization group”
- “What is data synchronization?” on page 80
- “Identifying the group” on page 80
- “Enabling the group to access data and files” on page 81
- “What is a subscription set?” on page 82
- “Defining a data filter for a group” on page 82
- “What is a data filter?” on page 83
- “Data filter example” on page 83
- “Assigning a user to a group” on page 84
- “What is a user?” on page 85

Creating a data synchronization group

Procedure:

1. Open the Create Group notebook. (See “Creating an object from the object tree” on page 70 for more information.)
2. Identify the group.
3. Associate subscription sets with the group to enable its members to access the data and files that they need.
4. Optional: Filter the source data available to the group, whether to improve performance or for security reasons.
5. Assign users to the group.
6. Click **OK**.

Each of these steps is explained in more detail in this section. After you create the group, enable it for synchronization so that the group’s members can synchronize data. For more information, see “Enabling synchronization” on page 129.

Related concepts:

- “What is data synchronization?” on page 80
- “What is a subscription set?” on page 82
- “Data filter example” on page 83
- “What is a data filter?” on page 83
- “What is a user?” on page 85

Related tasks:

- “Using the Mobile Devices Administration Center” on page 77
- “Identifying the group” on page 80
- “Enabling the group to access data and files” on page 81
- “Defining a data filter for a group” on page 82
- “Assigning a user to a group” on page 84

What is data synchronization?

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, you create replication subscriptions referencing the necessary data and files, collect these subscriptions into a subscription set, and assign the subscription set to the group.

Related concepts:

- "What is a subscription set?" on page 82
- "Data filter example" on page 83
- "What is a data filter?" on page 83
- "What is a user?" on page 85

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Creating a data synchronization group" on page 79
- "Identifying the group"
- "Enabling the group to access data and files" on page 81
- "Defining a data filter for a group" on page 82
- "Assigning a user to a group" on page 84

Identifying the group

When a user requests synchronization, the Sync Server authenticates the user, then determines the group to which the user belongs to begin replicating the information referenced in that group's subscriptions. For this reason, the 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. When you have completed these steps, return to Creating a data synchronization group.

Procedure:

1. Click the **Identification** tab of the Creating upload subscriptions notebook.
2. Fill in these fields:
   - **Name**

     Provide a descriptive, unique name for the group from 1 to 18 characters. Names are case sensitive.
   - **Description**

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

Related concepts:

- "What is data synchronization?"
- "What is a subscription set?" on page 82
- "Data filter example" on page 83
- "What is a data filter?" on page 83
- "What is a user?" on page 85

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Creating a data synchronization group" on page 79
- "Enabling the group to access data and files"
- "Defining a data filter for a group" on page 82
- "Assigning a user to a group" on page 84

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.

This task is part of the larger task of creating a data synchronization group. When you have completed these steps, return to Creating a data synchronization group.

Procedure:

1. Go to the Subscription Sets page of the Create Group notebook.
2. In the **Available subscription sets** list, select the subscription set that you want to assign. To select multiple subscription sets serially, hold down the Ctrl key while selecting; While selecting 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 set notebook now, or you can create a subscription set later and associate the group with it later. See "Making the subscription set available to a group" on page 127 for more information about creating a subscription set.
3. 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 Subscription set notebook. For more information, see "Making the subscription set available to a group" on page 127.

Related concepts:

- "What is data synchronization?" on page 80
- "What is a subscription set?" on page 82
- "Data filter example" on page 83
- "What is a data filter?" on page 83
- "What is a user?" on page 85

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Creating a data synchronization group" on page 79
- "Identifying the group" on page 80
- "Defining a data filter for a group" on page 82
- "Assigning a user to a group" on page 84

What is a subscription set?

A *subscription set* is a collection of replication subscriptions. A *replication subscription* is a specification for copying changed data or files from a source system to a target system at a specified frequency. It specifies the subset of data and files that members of a group are allowed to access as well as what SQL access privileges they have for the data.

Related concepts:

- "What is data synchronization?" on page 80
- "Data filter example" on page 83
- "What is a data filter?" on page 83
- "What is a user?" on page 85

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Creating a data synchronization group" on page 79
- "Identifying the group" on page 80
- "Enabling the group to access data and files" on page 81
- "Defining a data filter for a group"
- "Assigning a user to a group" on page 84

Defining a data filter for a group

Procedure:

1. On the Data Filter page of the Create Group notebook, click **Add**. The Add Data Filter window opens.
2. In the **Parameter name** field, type the name of the column in the table that you want to filter. The name is case sensitive.

   Recommendation: Uniquely identify your parameter names with additional text. For example, add a colon (:) to the beginning of the **Parameter name** .
3. In the **Default value** field, type the value that will serve as the default filter criteria.

   If you assign a data filter for an individual user in the group, the user filter values override this default. See "Filtering the data available to the user" on page 88 for more information about how to define a data filter for an individual user.
4. Click **OK**. The parameter-value pair is displayed on the Data Filter page.

Related concepts:

- "What is data synchronization?" on page 80
- "What is a subscription set?"
- "Data filter example" on page 83
- "What is a data filter?" on page 83
- "What is a user?" on page 85

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Creating a data synchronization group" on page 79

- “Identifying the group” on page 80
- “Enabling the group to access data and files” on page 81
- “Assigning a user to a group” on page 84

What is a data filter?

A *data filter* is a parameter-value pair specifying a table column and its value. The filter specifies which rows of a table will be included in the synchronization request. The column and its value substitute into the WHERE clause of the SELECT statement of the subscription that the group uses. You enter this WHERE clause when you create the subscription (see “Creating a DataPropagator subscription” on page 104 for more information). The WHERE clause is submitted to the source database during synchronization so that the DB2 Everyplace Sync Server returns only the subset of data required by a given group. More than one data filter can be applied for a specific user or group.

Related concepts:

- “What is data synchronization?” on page 80
- “What is a subscription set?” on page 82
- “Data filter example”
- “What is a user?” on page 85

Related tasks:

- “Using the Mobile Devices Administration Center” on page 77
- “Creating a data synchronization group” on page 79
- “Identifying the group” on page 80
- “Enabling the group to access data and files” on page 81
- “Defining a data filter for a group” on page 82
- “Assigning a user to a group” on page 84

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 replicated, data filters help improve synchronization performance and provide security for group-specific data.

For example, to set a data filter for East side hospital, the parameter name and value might be something like this:

```
Parameter Name = :Hospital
Default Value = Eastside
```

The data filter for Foothill hospital on the west side might look something like this:

```
Parameter Name = :Hospital
Default Value = Foothill
```

You can also set data filters for individual users by using a DB2 variable instead of a hard-coded value for the parameter name. You then enter a data filter value when you create each user. In this case, the group filter might look something like this:

```
Parameter Name = :Hospital
Default Value = None
```

Then you would assign values for the individual user filters as explained in “Filtering the data available to the user” on page 88.

Related concepts:

- “Using the Mobile Devices Administration Center” on page 77
- “What is a subscription set?” on page 82
- “What is a data filter?” on page 83
- “What is a user?” on page 85

Related tasks:

- “Creating a data synchronization group” on page 79
- “Identifying the group” on page 80
- “Enabling the group to access data and files” on page 81
- “Defining a data filter for a group” on page 82
- “Assigning a user to a group”

Assigning a user to a group

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.

Procedure:

1. Go to the Users page of the Create Group notebook.
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 notebook now, or you can create a user later and assign the user to a group later. For information on creating a user, see “Creating users in the Mobile Devices Administration Center” on page 85.
3. Click **>** to assign the users. Assigned users are displayed in the **Selected users** list. Click **>>** to assign all available users at once.

You can also assign users to a group from the Users notebook. See “Assigning the user to a group” on page 87 for more information.

Related concepts:

- “What is data synchronization?” on page 80
- “What is a subscription set?” on page 82
- “Data filter example” on page 83
- “What is a data filter?” on page 83
- “What is a user?” on page 85

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Creating a data synchronization group" on page 79
- "Identifying the group" on page 80
- "Enabling the group to access data and files" on page 81
- "Defining a data filter for a group" on page 82

What is a user?

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 the administration control database. The 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 form 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.

Related concepts:

- "What is data synchronization?" on page 80
- "What is a subscription set?" on page 82
- "Data filter example" on page 83
- "What is a data filter?" on page 83

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Creating a data synchronization group" on page 79
- "Identifying the group" on page 80
- "Enabling the group to access data and files" on page 81
- "Defining a data filter for a group" on page 82
- "Assigning a user to a group" on page 84

Defining mobile users to the Sync Server

This section provides information about how to define mobile users to the Sync Server using the Mobile Devices Administration Center. The topics covered are:

- "Creating users in the Mobile Devices Administration Center"
- "Identifying the user" on page 86
- "Assigning the user to a group" on page 87
- "Setting passwords for the user" on page 87
- "Filtering the data available to the user" on page 88
- "Registering a device" on page 88
- "Importing user definitions from another source" on page 89

Creating users in the Mobile Devices Administration Center

You can define users to the Sync Server by creating users in the Mobile Devices Administration Center (see "Creating users in the Mobile Devices Administration

Center" on page 85) or importing existing definitions directly into the administration control database (see "Importing user definitions from another source" on page 89).

Procedure:

1. Open the Create User notebook.
2. "Identifying the user"Identify the user.
3. "Assigning the user to a group" on page 87Assign the user to a group.
4. "Setting passwords for the user" on page 87Optional: Provide a password for the user.
5. "Filtering the data available to the user" on page 88Optional: Filter the source data available to the user. The data filter that you specify for the user overrides the filter that you set for the group.
6. Click **OK**.

Each step is explained further in this section.

After you create the user, the user must register his or her device as explained in "Registering a device" on page 88.

Related concepts:

- "What is a user?" on page 85

Related tasks:

- "Identifying the user"
- "Assigning the user to a group" on page 87
- "Setting passwords for the user" on page 87
- "Filtering the data available to the user" on page 88
- "Registering a device" on page 88
- "Importing user definitions from another source" on page 89

Identifying the user

For authentication purposes, the Sync Server must be able to uniquely identify each user by name.

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.

Procedure:

1. Click the **Identification** tab of the Creating upload subscriptions notebook.
2. Fill in these 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**

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

Related concepts:

- "What is a user?" on page 85

Related tasks:

- “Creating users in the Mobile Devices Administration Center” on page 85
- “Assigning the user to a group”
- “Setting passwords for the user”
- “Filtering the data available to the user” on page 88
- “Registering a device” on page 88
- “Importing user definitions from another source” on page 89

Assigning the user to a group

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 (as explained in “Registering a device” on page 88), but all other data synchronization fails. A user can belong to only one group at a time.

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 Administrator.

Procedure:

1. On the Identification page of the Create User notebook, 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 will need to create a group. You can click **Create** to open the Create Group notebook now, or you can create a group later and associate the group with a user later. See “Creating a data synchronization group” on page 79 for more information on creating a group.

You can also assign users to a group from the Group notebook. For more information, see“Assigning a user to a group” on page 84.

Related concepts:

- “What is a user?” on page 85

Related tasks:

- “Creating users in the Mobile Devices Administration Center” on page 85
- “Identifying the user” on page 86
- “Setting passwords for the user”
- “Filtering the data available to the user” on page 88
- “Registering a device” on page 88
- “Importing user definitions from another source” on page 89

Setting passwords for the user

For enhanced security and especially if encryption is enabled, 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 Administrator.

Procedure:

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

Related concepts:

- "What is a user?" on page 85

Related tasks:

- "Creating users in the Mobile Devices Administration Center" on page 85
- "Identifying the user" on page 86
- "Assigning the user to a group" on page 87
- "Filtering the data available to the user"
- "Registering a device"
- "Importing user definitions from another source" on page 89

Filtering the data available to the user

You can set a data filter value for individual users that overrides the value of the data filter that you set for the group.

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.

Procedure:

1. Ensure that the group filter specifies a DB2 variable for the parameter name instead of a hard-coded value. (For more information about group data filters, see "Defining a data filter for a group" on page 82.)
2. Go to the Data Filter page of the Create User notebook. The filters displayed here are those that you set for the group to which the user is assigned. Select the filter whose value you want to change.
3. Click **Change**. The Change Data Filter window opens.
4. In the **User override** field, type the value of the parameter for this user.
5. Click **OK**.

Related concepts:

- "What is a user?" on page 85

Related tasks:

- "Creating users in the Mobile Devices Administration Center" on page 85
- "Identifying the user" on page 86
- "Assigning the user to a group" on page 87
- "Setting passwords for the user" on page 87
- "Registering a device"
- "Importing user definitions from another source" on page 89

Registering a device

A users device is registered during the first synchronization of the device. The Sync Server will allow 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 behavior is to allow only a single device to register for a user. Once the property is changed to allow multiple devices for a user, the setting should not be changed back. If the setting is changed from multiple device to single device per user, the behavior is undefined and unsupported.

If a user has already been created and assigned to an enabled group, then this user will be automatically be enabled 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 values 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 reset the user. When a user is reset all information regarding a users device will be deleted from the control database. If a user tries to synchronize a new device before the administrator resets the user for the device, the new device will not be registered.

   Note: Be sure to synchronize the device before you reset it. Failure to so, will result in loss of new data that has not been synchronized.
2. AllowMultipleDevicesPerUser=true

   Allows a user to synchronize an unlimited number of devices without requiring the administrator to reset the user 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 reset individual devices. To reset all devices for a user, the administrator can simply reset the user.

   Note: If you reset the user, you must reset all the devices associated with that user. If you want to reset a particular device, you must edit the user.

Related concepts:

- "What is a user?" on page 85

Related tasks:

- "Creating users in the Mobile Devices Administration Center" on page 85
- "Identifying the user" on page 86
- "Assigning the user to a group" on page 87
- "Setting passwords for the user" on page 87
- "Filtering the data available to the user" on page 88
- "Importing user definitions from another source"

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 this information with the XML Scripting Tool, use the following command.

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

For more information see “XML Scripting tool DTD” on page 205 and “Utilizing the XML Scripting tool” on page 131.

Related concepts:

- “What is a user?” on page 85

Related tasks:

- “Creating users in the Mobile Devices Administration Center” on page 85
- “Identifying the user” on page 86
- “Assigning the user to a group” on page 87
- “Setting passwords for the user” on page 87
- “Filtering the data available to the user” on page 88
- “Registering a device” on page 88

Setting up data and replication sets

This section provides information about how to set up data and replication sets using the Mobile Devices Administration Center. The topics covered are:

- “Data set up and replication”
- “What is a replication subscription?” on page 91
- “What a subscription set?” on page 92
- “Creating a JDBC subscription” on page 93
- “Creating a DataPropagator subscription” on page 104
- “Creating an upload subscription” on page 112
- “Creating a file subscription” on page 118
- “Creating a custom subscription” on page 122

Data set up 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: DataPropagator subscriptions, JDBC subscriptions, upload subscriptions, file subscriptions, and custom subscriptions. To ensure that the data in a database on your source system can be synchronized with the target system and to determine what a particular source data type is mapped to on a target system, see “Data type mappings between DB2 Everyplace and data sources” on page 220.

- *JDBC subscriptions* provide users with access to data in source tables on a data source with a JDBC interface, including Oracle, DB2, Microsoft SQL Server, Informix, Sybase, and Lotus Domino..
- *DataPropagator subscriptions* provide users with access to data in source tables on a DB2 server.

- *Upload subscriptions* only allow the user to directly insert rows into a table on a source database. The source table may 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 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 custom interactions with the Sync Server and data sources.

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

Related concepts:

- "Replication Invocation" on page 102

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Creating a JDBC subscription" on page 93
- "Identifying the JDBC subscription" on page 94
- "Specifying the encryption level" on page 95
- "Specifying the source database driver and URL" on page 95
- "Specifying the mirror database driver and URL" on page 97
- "Specifying source tables, target tables, and privileges for the subscription" on page 98
- "Filtering the data that will be replicated during synchronization" on page 99
- "Batching synchronization requests" on page 101
- "Assigning a subscription set to a JDBC subscription" on page 103

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 Sync Server. Replication subscriptions include:

- JDBC subscriptions
- DataPropagator subscriptions

Related concepts:

- "Replication Invocation" on page 102

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Creating a JDBC subscription" on page 93
- "Identifying the JDBC subscription" on page 94
- "Specifying the encryption level" on page 95
- "Specifying the source database driver and URL" on page 95
- "Specifying the mirror database driver and URL" on page 97
- "Specifying source tables, target tables, and privileges for the subscription" on page 98

- “Filtering the data that will be replicated during synchronization” on page 99
- “Batching synchronization requests” on page 101
- “Assigning a subscription set to a JDBC subscription” on page 103

What a subscription set?

A *subscription set* is a container for replication subscriptions. You assign a subscription set to a group so that its members have access to the data and files defined in the replication subscriptions that are part of the subscription set. The subscriptions that you assign can be DataPropagator subscriptions, JDBC subscriptions, file subscriptions, or custom subscriptions. Each subscription set can contain an unlimited number of subscriptions.

When users start the synchronization User software on the device, they choose which subscription sets to synchronize. This menu of subscription sets is created from the list of subscription sets associated with the User’s group.

To enable a group of mobile users to access data and files that its members require, associate the subscription set with replication subscriptions that define tables and files as data sources available for replication.

For example, suppose your group of visiting nurses relies on an application called vnurse to perform their jobs. The vnurse application accesses patient and scheduling information in the PATIENTS and SCHEDULES tables, which reside on an enterprise 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 application (`vnurse`).

After defining the subscriptions, associate the subscriptions with an subscription set and then assign that subscription set to the VNURSE group to enable the visiting nurses to replicate the data that they need. When the visiting nurses synchronize their data, they receive the latest copy of the resource files, as well as updates to the PATIENTS and SCHEDULES tables.

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

Related concepts:

- “Replication Invocation” on page 102

Related tasks:

- “Using the Mobile Devices Administration Center” on page 77
- “Creating a JDBC subscription” on page 93
- “Identifying the JDBC subscription” on page 94
- “Specifying the encryption level” on page 95
- “Specifying the source database driver and URL” on page 95
- “Specifying the mirror database driver and URL” on page 97
- “Specifying source tables, target tables, and privileges for the subscription” on page 98
- “Filtering the data that will be replicated during synchronization” on page 99
- “Batching synchronization requests” on page 101

- “Assigning a subscription set to a JDBC subscription” on page 103

Creating a JDBC subscription

This section provides information about how to create a JDBC subscription using the Mobile Devices Administration Center. The topics covered are:

- “Creating a JDBC subscription”
- “What is a JDBC subscription?”
- “Identifying the JDBC subscription” on page 94
- “Specifying the encryption level” on page 95
- “Specifying the source database driver and URL” on page 95
- “Specifying the mirror database driver and URL” on page 97
- “Specifying source tables, target tables, and privileges for the subscription” on page 98
- “Filtering the data that will be replicated during synchronization” on page 99
- “Batching synchronization requests” on page 101
- “Replication Invocation” on page 102
- “Assigning a subscription set to a JDBC subscription” on page 103

Creating a JDBC subscription

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

1. Open the Create JDBC Subscription notebook.
2. Identifying the subscription.
3. Specifying the encryption level.
4. Specifying the source database driver and URL.
5. Specifying the mirror database driver and URL.
6. Specifying source tables, target tables, and access privileges for the subscription.
7. Specifying how to filter the data in the source table.
8. Assigning subscription sets to the JDBC subscription.
9. Click **OK**.

Related concepts:

- “Data set up and replication” on page 90
- “Replication Invocation” on page 102

What is a JDBC subscription?

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 synchronization.

Related concepts:

- “Data set up and replication” on page 90
- “Replication Invocation” on page 102

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Identifying the JDBC subscription"
- "Specifying the encryption level" on page 95
- "Specifying the source database driver and URL" on page 95
- "Specifying the mirror database driver and URL" on page 97
- "Specifying source tables, target tables, and privileges for the subscription" on page 98
- "Filtering the data that will be replicated during synchronization" on page 99
- "Batching synchronization requests" on page 101
- "Assigning a subscription set to a JDBC subscription" on page 103

Identifying the JDBC subscription

After authenticating a user, the 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 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.

Procedure:

1. Click the **Identification** tab of the Creating upload subscriptions notebook.
2. Fill in these 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 set up and replication" on page 90
- "Replication Invocation" on page 102

Related concepts:

- "Data set up and replication" on page 90
- "Replication Invocation" on page 102

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Identifying the JDBC subscription"
- "Specifying the encryption level" on page 95
- "Specifying the source database driver and URL" on page 95
- "Specifying the mirror database driver and URL" on page 97
- "Specifying source tables, target tables, and privileges for the subscription" on page 98
- "Filtering the data that will be replicated during synchronization" on page 99
- "Batching synchronization requests" on page 101
- "Assigning a subscription set to a JDBC subscription" on page 103

Specifying the encryption level

Encryption support for Windows CE and Win32 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 Win32 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.

Procedure:

To specify the encryption level 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 concepts:

- "Data set up and replication" on page 90
- "Replication Invocation" on page 102

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Creating a JDBC subscription" on page 93
- "Identifying the JDBC subscription" on page 94
- "Specifying the source database driver and URL"
- "Specifying the mirror database driver and URL" on page 97
- "Specifying source tables, target tables, and privileges for the subscription" on page 98
- "Filtering the data that will be replicated during synchronization" on page 99
- "Batching synchronization requests" on page 101
- "Assigning a subscription set to a JDBC subscription" on page 103

Specifying the source database driver and URL

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.

Prerequisites:

You must specify the source and mirror databases. For *upload* subscriptions, no mirror database is required.

Procedure:

On the Source page of the Create JDBC Subscription notebook, 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 IBM DB2 UDB local is jdbc:db2:*{database}*. Replace *{database}* with the name of the DB2 database you plan to use. For example, for Microsoft SQL Server example, you can use:

```
jdbc:microsoft:sqlserver://myhost.mycompany.com:1433;SelectMethod=cursor,
     DatabaseName=myDatabase
```

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.

Note: 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.

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 Sync Server subscription, you cannot edit this value. Control Table Space field is only valid for DB2 source.

Note for OS390 source: You can enter database_name.tablespace in a text entry field. If you only specify one name (no ″dot″ in between), DB2 Everyplace will consider it is a table space name in the default database.

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

Related concepts:

- "Data set up and replication" on page 90
- "Replication Invocation" on page 102

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Creating a JDBC subscription" on page 93
- "Identifying the JDBC subscription" on page 94
- "Specifying the encryption level" on page 95

- “Specifying the mirror database driver and URL”
- “Specifying source tables, target tables, and privileges for the subscription” on page 98
- “Filtering the data that will be replicated during synchronization” on page 99
- “Batching synchronization requests” on page 101
- “Assigning a subscription set to a JDBC subscription” on page 103

Specifying the mirror database driver and URL

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.

Prerequisites:

You need to specify the mirror database for a *synchronizing* subscription, but not for a *upload* subscription.

Procedure:

On the Mirror page of the Create JDBC Subscription notebook, 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 UDB 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.

Click Test connection to verify the database settings.

Related concepts:

- “Data set up and replication” on page 90
- “Replication Invocation” on page 102

Related tasks:

- “Using the Mobile Devices Administration Center” on page 77
- “Creating a JDBC subscription” on page 93
- “Identifying the JDBC subscription” on page 94
- “Specifying the encryption level” on page 95
- “Specifying the source database driver and URL” on page 95
- “Specifying source tables, target tables, and privileges for the subscription” on page 98
- “Filtering the data that will be replicated during synchronization” on page 99
- “Batching synchronization requests” on page 101
- “Assigning a subscription set to a JDBC subscription” on page 103

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.

You must 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.

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.

Procedure:

To add a table to the subscription:

1. On the Identification page of the JDBC Subscription notebook, 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.

   Click **Filter** to change the source filter used to limit the available tables in the table list. The Filter window opens.

   The % symbol is a wildcard. Typing % in the Source schema and Source table fields displays all available tables in the table list.
4. For Lotus Domino, click **Advanced** on the Add Table window to select a primary key for the appropriate columns. For more information see, “Configuring the JDBC driver to access Lotus Domino data” on page 28.
5. 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.
6. 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.
7. 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 only to view the data, 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 Universal Database *Administration Guide: Design and Implementation*.
8. Optional: Select the **Encrypt on device** checkbox. This enables encryption on the mobile device database. (Encrytion is only available on WinCE, Win32, and Palm OS client platforms).
9. 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.
10. 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.

11. 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 only available for DB2 sources. DB2 on iSeries is not supported.
12. 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.
13. Click **Add** to add the replication source to the subscription. Click **Close** to close the Add Table window.

Related concepts:

- "Data set up and replication" on page 90
- "Replication Invocation" on page 102

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Creating a JDBC subscription" on page 93
- "Identifying the JDBC subscription" on page 94
- "Specifying the encryption level" on page 95
- "Specifying the source database driver and URL" on page 95
- "Specifying the mirror database driver and URL" on page 97
- "Filtering the data that will be replicated during synchronization"
- "Batching synchronization requests" on page 101
- "Assigning a subscription set to a JDBC subscription" on page 103

Filtering the data that will be replicated during synchronization

When you create a subscription to a particular table (the PATIENTS table for your group of visiting nurses, for example), performance considerations might force you to replicate only a subset of the data in the table. You specify this subset by choosing individual rows or columns to be replicated for each table in the subscription.

For example, you might decide that the only columns relevant for visiting nurses using the PATIENTS table are the PATIENT_NAME, ADDRESS, PHONE, and DOCTOR fields. You might then decide to further filter the data by defining a WHERE clause that filters data for nurses working for a particular doctor.

You can combine data filters for the subscription with filters set for the group or for individual users to pare down the data even further.

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.

Procedure:

To filter data that will be replicated during synchronization:

1. From the Define Replication Subscription window, click **Advanced**. The Advanced Subscription Definition notebook opens.

Note: The source table primary key columns must be subscribed. If the source table has columns which are not part of the primary key, at least one of those columns must be subscribed.

2. Optional: Select the columns to use in the subscription.
   a. Go to the Target Columns page of the Advanced Subscription Definition notebook.
   b. Select the columns to create on the mobile device using the Subscribe checkbox. Columns not selected will not be created on the mobile device.
   c. Select the columns to replicate to the mirror database using the Replicate checkbox. You may need to replicate a column to the mirror database, but not want to create it on the handheld. For example, if a NOT NULLABLE column without a DEFAULT VALUE exists on the server and you are not creating it on the mobile device, you must replicate it to the mirror database and supply a default value using the Default value data filter. The Sync Server will replicate any rows created by the mobile device to the source using the DEFAULT VALUE specified in the Default value data filter to complete the contents of the NOT NULLABLE source column that does not have a default value on the server.
   d. Click **Change** to modify a target column name, Default value data filter, or constraint.
   e. Click **Close** to close the Create Index window.
3. Optional: Define indexes for the target table.
   a. Go to the Indexes page of the Advanced Subscription Definition notebook.
   b. Click Add to open the Create Index window. The Create Index window opens.
   c. Type the name index in the Name field.
   d. Select the columns you want to include in the index using the **>, >>, <**, and **<<** push buttons.
   e. Order the index using the **Move Up** and **Move Down** push buttons.
   f. Select **Ascending** or **Descending** index.
   g. Click **Add** to add the index.
   h. Click **Close** to close the Create Index window.
4. Optional: Define a WHERE clause that selects individual rows to be included in the subscription. You might decide that you want only certain rows to be part of the subscription even if all columns appear in the replica. To display sample WHERE clauses, click **Examples**.
   a. Go to the Rows page of the Advanced Subscription Definition window.
   b. In the **All rows needed** field, type a WHERE clause that defines the selection of rows that you want to appear in the subscription. You can only select rows from the table that you selected in Step 1.

      For example, suppose you want your visiting nurses to see only the rows of data in the PATIENTS table where the value of the JOBCODE column is VNURSE. You would enter the following WHERE clause:

      ```
      JOBCODE='VNURSE'
      ```

      It is not necessary to type WHERE at the beginning of the clause.
   c. In the **Subset of rows for individual users** field, you can type a clause similar to a SQL WHERE clause to further limit the rows that an individual user can see. Ensure you do not type ″WHERE″ at the beginning of the clause. You might use a parameter, whose value varies by the user. For

detailed information about defining a filter at the group level or user level, see “Filtering data at the subscription level” on page 43.

For example, suppose that you want users to see only data related to their own job codes. Instead of hard-coding a value for the JOBCODE parameter for the entire subscription, define a variable that references the data filter value defined for each individual user:

```
JOBCODE=':JOB'
```

In the data filters for individual users, you could then set the :JOB parameter to each user’s job code. See “Filtering the data available to the user” on page 88 and “Filtering data at the subscription level” on page 43 for more information on filtering data for individual users.

d. Click **OK** to close the Advanced Subscription Definition window.

5. Click **OK** to return to the Define Replication Subscription window.

Note: A series of changes made to a single row in a DB2 Everyplace table on the user’s device between synchronizations will be communicated to the Sync Server as a single cumulative change. As a result, a DELETE operation followed by an INSERT operation for the same primary key will be communicated to the Sync Server as an UPDATE operation if the operations occur with no intervening synchronizations. If they occur with an intervening synchronization, then they are communicated separately. Doing a series of changes to a single row can lead to an inconsistency because the subscription does not transfer all columns between the source table and the user table. An UPDATE operation will retain the current values in the unsubscribed column(s), while the INSERT will set the unsubscribed columns to default values.

Related concepts:

- “Data set up and replication” on page 90
- “Replication Invocation” on page 102

Related tasks:

- “Using the Mobile Devices Administration Center” on page 77
- “Creating a JDBC subscription” on page 93
- “Identifying the JDBC subscription” on page 94
- “Specifying the encryption level” on page 95
- “Specifying the source database driver and URL” on page 95
- “Specifying the mirror database driver and URL” on page 97
- “Specifying source tables, target tables, and privileges for the subscription” on page 98
- “Batching synchronization requests”
- “Assigning a subscription set to a JDBC subscription” on page 103

Batching synchronization 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.

Procedure:

1. Ensure a source and mirror database appear on the Identification page of the Create JDBC subscription notebook. Then click **Define Subscription**. The Define Replication Subscription window opens.

2. Click **Timing**. The Subscription Timing window opens.
3. In the **Batch window** field, specify a number of seconds that you want the Sync Server to wait between replication cycles. You can also specify 0, and initiate replication manually, via dsyreplicate.bat.
4. Click **OK**.

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.

Related concepts:

- "Data set up and replication" on page 90
- "Replication Invocation"

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Creating a JDBC subscription" on page 93
- "Identifying the JDBC subscription" on page 94
- "Specifying the encryption level" on page 95
- "Specifying the source database driver and URL" on page 95
- "Specifying the mirror database driver and URL" on page 97
- "Specifying source tables, target tables, and privileges for the subscription" on page 98
- "Filtering the data that will be replicated during synchronization" on page 99
- "Assigning a subscription set to a JDBC subscription" on page 103

Replication Invocation

For performance reasons, the Sync Server does not immediately replicate changes received from the synchronization requests to the source, and vice versa. Replication invocation is directed by one of the following:

- Batching of synchronization requests by a fixed elapsed period of time.
- Configuring a subscription to perform replication at any time you issue a special command.

When batching synchronization requests, the Sync Server begins replicating synchronization requests when the number of seconds defined for the batch window has elapsed since the last replication finished.

For JDBC subscriptions, when you have two devices which both get data from the same subscription, it will take two replication cycles on the server before new data entered in the first device can be synchronized to the second device. The flow of the data goes as follows:

- Data is entered on device D1.
- Device D1 synchronizes, data goes to the mirror database.
- The Sync Server performs the first replication cycle and the data is on the source database.
- Device D2 synchronizes, but does not yet receives the data originally entered on device D1.
- Sync Server performs the second replication cycle.
- Device D2 synchronizes after the second replication cycle, and now D2 receives the data originally entered on device D1.

For information about how to set the Sync Server to perform replication at any given time, see “Replicating on demand” on page 40.

Related concepts:

- “Data set up and replication” on page 90

Related tasks:

- “Using the Mobile Devices Administration Center” on page 77
- “Creating a JDBC subscription” on page 93
- “Identifying the JDBC subscription” on page 94
- “Specifying the encryption level” on page 95
- “Specifying the source database driver and URL” on page 95
- “Specifying the mirror database driver and URL” on page 97
- “Specifying source tables, target tables, and privileges for the subscription” on page 98
- “Filtering the data that will be replicated during synchronization” on page 99
- “Batching synchronization requests” on page 101
- “Assigning a subscription set to a JDBC subscription”

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.

Procedure:

To specify the subscription sets that use this subscription:

1. Go to the subscription sets page of the Create JDBC Subscription notebook.
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 notebook. The Create Subscription Set notebook 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 set up and replication” on page 90
- “Replication Invocation” on page 102

Related tasks:

- “Using the Mobile Devices Administration Center” on page 77
- “Creating a JDBC subscription” on page 93
- “Identifying the JDBC subscription” on page 94
- “Specifying the encryption level” on page 95
- “Specifying the source database driver and URL” on page 95
- “Specifying the mirror database driver and URL” on page 97
- “Specifying source tables, target tables, and privileges for the subscription” on page 98
- “Filtering the data that will be replicated during synchronization” on page 99
- “Batching synchronization requests” on page 101

Creating a DataPropagator subscription

This section provides information about how to create a DataPropagator subscription using the Mobile Devices Administration Center. The topics covered are:

- “Creating a DataPropagator subscription”
- “Identifying the subscription” on page 105
- “Specifying the encryption level” on page 106
- “Mapping the source to the target” on page 107
- “Identifying the mirror database” on page 108
- “Adding a replication source” on page 108
- “Creating a filter for replicated data” on page 109
- “Data filtering and replication” on page 111
- “Providing a user ID and password for the subscription” on page 111

Creating a DataPropagator subscription

A DataPropagator subscription provides specifications for how data in a source table is to be synchronized to a table on the mobile device. Each subscription that you create can reference one or more tables.

Restrictions:

For DataPropagator subscriptions, the mirror database must be located on the same server as the replication engine. The DataPropagator Capture program is not able to capture any remote databases.

Prerequisites:

Ensure that the table is defined in DB2 Universal Database as a replication source, and that you defined at least one NOT nullable, primary key column for the table. (For information on defining replication sources, see *DB2 Universal Database Replication Guide and Reference*).

Procedure:

To create a DataPropagator subscription:

1. Open the Create DataPropagator subscription notebook in the Mobile Devices Administration Center.
2. Identify the subscription.
3. Specify the encryption level.
4. Map the source database on the enterprise system to its target database on the mobile device.
5. Identify the mirror database. If you set up synchronization in a three-tier structure, the mirror is on the mid-tier system.
6. Add source tables to the subscription and provide information about what SQL privileges you will allow for those tables.
7. Optional: Specify how to filter the data in the source table.

   For example, do you want all data in the source table to be replicated in every synchronization? Only certain rows? The data filtering you specify for the subscription applies to all groups that use the subscription.
8. Specify how you want to batch synchronization requests by defining a period of time to elapse between replication processes.
9. Provide the user ID and password that will be used to access the source system on the group's behalf.
10. Click **OK**.

Related concepts:

- "What is a replication source?" on page 32
- "Data filtering and replication" on page 111
- "Improving replication performance" on page 39

Identifying the subscription

After authenticating a user, the 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 Sync Server must be able to uniquely identify each subscription by name.

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

Procedure:

1. Click the **Identification** tab of the Creating upload subscriptions notebook.
2. Fill in these fields:
   - **Name**Provide a descriptive, unique name for the subscription from 1 to 18 characters long.

     For DB2 DataPropagator subscriptions, the system converts the name that you specify here to uppercase upon saving it in the administration control database. After the name is saved in the control database, you cannot rename the subscription because this function is not supported by DB2 DataPropagator. You must first delete the existing subscription, then create a new subscription with the new name.
   - **Description**

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

Related concepts:

- "What is a replication source?" on page 32
- "Data filtering and replication" on page 111

- “Improving replication performance” on page 39

Related tasks:

- “Using the Mobile Devices Administration Center” on page 77
- “Creating a DataPropagator subscription” on page 104
- “Specifying the encryption level”
- “Mapping the source to the target” on page 107
- “Identifying the mirror database” on page 108
- “Adding a replication source” on page 108
- “Creating a filter for replicated data” on page 109
- “Batching synchronization requests” on page 40
- “Providing a user ID and password for the subscription” on page 111

Specifying the encryption level

Encryption support for Windows CE and Win32 platforms is only available with the Microsoft High Encryption Pack available from Microsoft.

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

Prerequisites:

Encryption support on Palm OS requires Palm OS 3.3 or later. The Microsoft High Encryption Pack must be installed to use encryption between Windows CE/Pocket PC clients or Win32 platform clients and the DB2 Everyplace Sync Server.

Procedure:

To specify the encryption level:

On the Identification page, select the encryption level using the **Encryption level** field. After the encryption level is modified, all users subscribing to that subscription must be reset. 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 a replication source?” on page 32
- “Data filtering and replication” on page 111
- “Improving replication performance” on page 39

Related tasks:

- “Using the Mobile Devices Administration Center” on page 77
- “Creating a DataPropagator subscription” on page 104
- “Identifying the subscription” on page 105

- “Mapping the source to the target”
- “Identifying the mirror database” on page 108
- “Adding a replication source” on page 108
- “Creating a filter for replicated data” on page 109
- “Batching synchronization requests” on page 40
- “Providing a user ID and password for the subscription” on page 111

Mapping the source to the target

The *source database* is the database serving as the source for data replication.

The *target database* is the database on the mobile device receiving replicated data from a source system.

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

Procedure:

To map the source database on the enterprise system to its target database on the mobile device, provide information in the following fields on the Identification page of the Create DataPropagator subscription notebook:

Source database

The source database must be cataloged. Choose a name from a list of cataloged databases by clicking the ... push button.

Target database

This field automatically defaults to the name that you typed in the **Source database** field. You can override this target database name only if this is the first subscription defined against this source database.

This field does not apply to devices running Palm OS, because that operating system does not use a traditional database structure.

Related concepts:

- “What is a replication source?” on page 32
- “Data filtering and replication” on page 111
- “Improving replication performance” on page 39

Related tasks:

- “Using the Mobile Devices Administration Center” on page 77
- “Creating a DataPropagator subscription” on page 104
- “Identifying the subscription” on page 105
- “Specifying the encryption level” on page 106
- “Identifying the mirror database” on page 108
- “Adding a replication source” on page 108
- “Creating a filter for replicated data” on page 109
- “Batching synchronization requests” on page 40
- “Providing a user ID and password for the subscription” on page 111

Identifying the mirror database

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

Prerequisites:

Before you begin setting up synchronization, you must create the mirror database on the mid-tier system. You can create the mirror database by clicking the **Create** button to launch the Create Database wizard. You can then identify the mirror database in the subscription.

Procedure:

To identify the mirror database, choose a mirror database name from the list in the **Mirror database** field on the Identification page of the Create DataPropagator subscription notebook. This field displays all databases defined on the server where you invoked the Mobile Devices Administration Center.

Related concepts:

- “What is a replication source?” on page 32
- “Data filtering and replication” on page 111
- “Improving replication performance” on page 39

Related tasks:

- “Using the Mobile Devices Administration Center” on page 77
- “Creating a DataPropagator subscription” on page 104
- “Identifying the subscription” on page 105
- “Specifying the encryption level” on page 106
- “Mapping the source to the target” on page 107
- “Adding a replication source”
- “Creating a filter for replicated data” on page 109
- “Batching synchronization requests” on page 40
- “Providing a user ID and password for the subscription” on page 111

Adding a replication source

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

Procedure:

1. Ensure that you defined the table as a DB2 DataPropagator replication source. You can use the DB2 Control Center to define the replication source. For more information, see the *DB2 Universal Database Replication Guide and Reference*.

   Because replication sources can be used and managed by tools other than the DB2 Everyplace Sync Server, when you define the source ensure that you control access to the data in a way that all applications can take advantage of.
2. On the Identification page of the Create DataPropagator subscription notebook, click the

   ...

   push button to choose a source database from a list.
3. Click **Define Subscription**. The Define Replication Subscription window opens.

The **Subscription name** field reflects the name of the subscription that you entered on the Identification page. The **Apply qualifier** field represents a unique qualifier that the Sync Server uses for all subscriptions defined for this database.

The Apply qualifier streamlines the process of synchronizing the mid-tier mirror database with its master database on the enterprise system; when the Apply program runs, DataPropagator replicates data for all subscriptions with the same Apply qualifier. The Apply qualifier defaults to the name of the mirror database.

4. In the Define Replication Subscription window, click **Add**. The Add window opens.
5. In the **Replication source** list, select the replication sources to add to this subscription.
6. In the **Access privileges** field, 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 training nurses are allowed only to view the data, you need to create two different subscriptions to handle this difference in access privileges. These subscriptions must use separate mirror tables. For more information about SQL access privileges, see the *DB2 Universal Database SQL Reference*.

   After you add a replication source to the subscription, you can change only the SQL access privileges defined for the source.
7. 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. The **Create target table** check box is always disabled.

   Within the same subscription, a source table can be mapped to more than one target table name. However, DB2 DataPropagator uses a single change data table to replicate changes to and from both target tables.
8. Click **Add** to add the replication source to the subscription.

Related concepts:

- "What is a replication source?" on page 32
- "Data filtering and replication" on page 111
- "Improving replication performance" on page 39

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Creating a DataPropagator subscription" on page 104
- "Identifying the subscription" on page 105
- "Specifying the encryption level" on page 106
- "Mapping the source to the target" on page 107
- "Identifying the mirror database" on page 108
- "Creating a filter for replicated data"
- "Batching synchronization requests" on page 40
- "Providing a user ID and password for the subscription" on page 111

Creating a filter for replicated data

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

Procedure:

1. In the Define Replication Subscription window, select the table for which you want to define column or row filtering.
2. Click **Advanced**. The Advanced Subscription Definition notebook opens.
3. Optional: Define a WHERE clause that selects individual rows to be included in the subscription. You might decide that you want only certain rows to be part of the subscription even if all columns appear in the replica.
   a. Go to the Rows page of the Advanced Subscription Definition window.
   b. In the **All rows needed** field, type a WHERE clause that defines the selection of rows that you want to appear in the subscription. You can only use the table that you selected in Step 1.

      For example, suppose you want your visiting nurses to see only the rows of data in the PATIENTS table where the value of the JOBCODE column is VNURSE. You could enter the following WHERE clause:

      ```
      JOBCODE='VNURSE'
      ```

      The WHERE keyword must not be used at the beginning of the clause.

      You can click **Examples** to display sample WHERE clauses.
   c. In the **Subset of rows for individual users** field, you can type a clause similar to a SQL WHERE clause to further limit the rows that an individual user can see. Ensure you do not type ″WHERE″ at the beginning of the clause. You might use a parameter, whose value varies by the user. For detailed information about defining a filter at the group level or user level, see “Filtering data at the subscription level” on page 43.

      For example, suppose that you want users to see only data related to their own job codes. Instead of hard-coding a value for the JOBCODE parameter for the entire subscription, define a variable that references the data filter value defined for each individual user:

      ```
      JOBCODE=':JOB'
      ```

      In the data filters for individual users, you could then set the :JOB parameter to each user’s job code. See “Filtering the data available to the user” on page 88 and “Filtering data at the subscription level” on page 43 for more information on filtering data for individual users.
   d. Click **OK** to close the Advanced Subscription Definition window.

Note: A series of changes made to a single row in a DB2 Everyplace table on the user’s device between synchronizations will be communicated to the Sync Server as a single cumulative change. As a result, a DELETE followed by an INSERT for the same primary key will be communicated to the SyncServer as an UPDATE, if the operations occur with no intervening synchronizations. If they occur with an intervening synchronization, then they are communicated separately. This can lead to an inconsistency when the subscription does not transfer all columns between the source table and the user table, as an UPDATE will retain the current values in the unsubscribed column(s), while the INSERT will set the unsubscribed columns to default values.

Related concepts:

- “What is a replication source?” on page 32
- “Data filtering and replication” on page 111
- “Improving replication performance” on page 39

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Creating a DataPropagator subscription" on page 104
- "Identifying the subscription" on page 105
- "Specifying the encryption level" on page 106
- "Mapping the source to the target" on page 107
- "Identifying the mirror database" on page 108
- "Adding a replication source" on page 108
- "Batching synchronization requests" on page 40
- "Providing a user ID and password for the subscription"

Data filtering and replication

When you create a subscription to a particular table (for example, the PATIENTS table for your group of visiting nurses), performance considerations might force you to replicate only a subset of the data in the subscription. You specify this subset by choosing individual rows to be replicated for each table in the subscription.

For example, you might decide that the only columns relevant for visiting nurses using the PATIENTS table are the PATIENT_NAME, ADDRESS, PHONE, and DOCTOR fields. You might then decide to further filter the data by defining a WHERE clause that filters data for nurses working for a particular doctor.

You can combine data filters for the subscription with filters set for the group or for individual users to pare down the data even further.

Related concepts:

- "What is a replication source?" on page 32
- "Improving replication performance" on page 39

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Creating a DataPropagator subscription" on page 104
- "Identifying the subscription" on page 105
- "Specifying the encryption level" on page 106
- "Mapping the source to the target" on page 107
- "Identifying the mirror database" on page 108
- "Adding a replication source" on page 108
- "Creating a filter for replicated data" on page 109
- "Batching synchronization requests" on page 40
- "Providing a user ID and password for the subscription"

Providing a user ID and password for the subscription

All subscriptions defined against a source database share the same user ID and password to gain access to the database. If this is the first subscription created against the source database, you need to specify this user ID and password. To do this, type the following values on the Authentications page of the Create DataPropagator subscription notebook:

- A valid DB2 Universal Database user ID. Ensure that 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.

- A valid DB2 Universal Database password for the user ID. Type the password again in the **Verify password** field.
  - For DataPropagator subscription, the user ID requires DBADM privileges.
  - For JDBC subscriptions, the user ID requires SYSADM or DBADM privileges.

For DataPropagator Subscriptions, the Sync Server creates a password file from this information that DataPropagator uses to access the source database. The password file is stored in the directory where the Sync Server runs.

Related concepts:

- "What is a replication source?" on page 32
- "Data filtering and replication" on page 111
- "Improving replication performance" on page 39

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Creating a DataPropagator subscription" on page 104
- "Identifying the subscription" on page 105
- "Specifying the encryption level" on page 106
- "Mapping the source to the target" on page 107
- "Identifying the mirror database" on page 108
- "Adding a replication source" on page 108
- "Creating a filter for replicated data" on page 109
- "Batching synchronization requests" on page 40

Creating an upload subscription

This section provides information about how to create an upload subscription using the Mobile Devices Administration Center. The topics covered are:

- "Creating an upload subscription"
- "What is an upload subscription?" on page 113
- "Identifying the upload subscription" on page 114
- "Specifying the encryption level" on page 115
- "Specifying the source database driver and URL" on page 115
- "Specifying source tables and target tables" on page 116
- "Assigning a subscription set to an upload subscription" on page 117

Creating an upload subscription

To create an upload subscription:

1. Open the Create Upload Subscription notebook.
2. Identifying the upload subscription.
3. Specifying the encryption level.
4. Specifying the source database driver and URL.
5. Specifying source tables and target tables for the subscription.
6. Assigning subscription sets to the upload subscription.
7. Click **OK**.

Related concepts:

- "What is an upload subscription?"

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Identifying the upload subscription" on page 114
- "Specifying the encryption level" on page 115
- "Specifying the source database driver and URL" on page 115
- "Specifying source tables and target tables" on page 116
- "Assigning a subscription set to an upload subscription" on page 117

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 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 Sync Server is alleviated.

In an upload subscription, when you synchronize with the 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 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 (MDAC) on the Sync Server. Other operations, like DELETE and UPDATE, are either ignored by the source database or rejected. Table 5 lists client operations and results.

Table 5. 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 MDAC

Table 5. Client's possible operations and their consequences (continued)

Operation	Results
Client's UPDATE operations	Rejected with error messages logged in the client and the MDAC
Client's DELETE operations	Ignored silently
First-time synchronization	Tables created in the client with no rows

Use the Create Upload Subscription notebook to create an upload table subscription. See "Creating an object from the object tree" on page 70 for information on how to open the notebook. Each table subscription that you create can reference one or more tables.

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Creating an upload subscription" on page 112
- "Identifying the upload subscription"
- "Specifying the encryption level" on page 115
- "Specifying the source database driver and URL" on page 115
- "Specifying source tables and target tables" on page 116
- "Assigning a subscription set to an upload subscription" on page 117

Identifying the upload subscription

After authenticating a user, the 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 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.

Procedure:

1. Click the **Identification** tab of the Creating upload subscriptions notebook.
2. Fill in these fields:
   - **Name**

     Type a descriptive, unique 1- to 18-character name for the upload subscription
   - **Description**

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

Related concepts:

- "What is an upload subscription?" on page 113

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Creating an upload subscription" on page 112
- "Specifying the encryption level" on page 115
- "Specifying the source database driver and URL" on page 115
- "Specifying source tables and target tables" on page 116
- "Assigning a subscription set to an upload subscription" on page 117

Specifying the encryption level

Encryption support for Windows CE and Win32 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 Win32 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.

Procedure:

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 113

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Creating an upload subscription" on page 112
- "Identifying the upload subscription" on page 114
- "Specifying the source database driver and URL"
- "Specifying source tables and target tables" on page 116
- "Assigning a subscription set to an upload subscription" on page 117

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.

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

Procedure:

On the **Source** page of the Create Upload Subscription notebook, 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 IBM DB2 UDB local is jdbc:db2:*{database}*. Replace *{database}* with the name of the DB2 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 113

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Creating an upload subscription" on page 112
- "Identifying the upload subscription" on page 114
- "Specifying the encryption level" on page 115
- "Specifying source tables and target tables"
- "Assigning a subscription set to an upload subscription" on page 117

Specifying source tables and target tables

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.

You must define at least one table for every upload subscription.

Procedure:

To add a table to the subscription:

1. On the **Identification** page of the Create Upload Subscription notebook, 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.

   Click **Filter ...** to change the source filter used to limit the available tables in the table list. The Filter window opens.

   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, by default the system uses the schema of the source table.
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 replication source to the subscription. Click **Close** to close the Add Table window.

Related concepts:

- "What is an upload subscription?" on page 113

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Creating an upload subscription" on page 112
- "Identifying the upload subscription" on page 114
- "Specifying the encryption level" on page 115
- "Specifying the source database driver and URL" on page 115
- "Assigning a subscription set to an upload subscription"

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.

Procedure:

1. Go to the subscription sets page of the Create Upload Subscription notebook.
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.

   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 notebook. The Create Subscription Set notebook 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:

- "What is an upload subscription?" on page 113

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Creating an upload subscription" on page 112
- "Identifying the upload subscription" on page 114
- "Specifying the encryption level" on page 115
- "Specifying the source database driver and URL" on page 115
- "Specifying source tables and target tables" on page 116

Creating a file subscription

This section provides information about how to create a file subscription using the Mobile Devices Administration Center. The topics covered are:

- "Creating a file subscription"
- "What is a file subscription?"
- "Identifying the subscription" on page 119
- "Specifying the encryption level" on page 119
- "Specifying the name of the source file" on page 120
- "Specifying subscription sets that use the subscription" on page 121
- "Flagging a subscription that references a changed file" on page 121

Creating a file subscription

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 one file. Thus, if your group of users needs access to sveral files, you must create several file subscriptions (then assign them to a subscription set as explained in"Specifying what information the subscription set accesses", and assign the subscription set to the group as explained in "Enabling the group to access data and files"). 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. 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 froma morning meeting.

To create a file subscription:

1. Open the Create File Subscription notebook. See "Creating synchronization objects."
2. Identify the subscription.
3. Specify the encryption level.
4. Specify the name of the source file.
5. Specify the name of a device type (do we need a separate topic for this?)
6. Specify the subscription sets that will use the subscription.
7. Click **OK**.

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Identifying the subscription" on page 119
- "Specifying the encryption level" on page 119
- "Specifying the name of the source file" on page 120
- "Specifying subscription sets that use the subscription" on page 121
- "Flagging a subscription that references a changed file" on page 121

What is a file subscription?

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:

- "Using the Mobile Devices Administration Center" on page 77
- "Identifying the subscription"
- "Specifying the encryption level"
- "Specifying the name of the source file" on page 120
- "Specifying subscription sets that use the subscription" on page 121
- "Flagging a subscription that references a changed file" on page 121

Identifying the subscription

After authenticating a user, the 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 Sync Server must be able to uniquely identify each file 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.

Procedure:

To identify a file subscription:

1. Click the **Identification** tab of the Creating upload subscriptions notebook.
2. Fill in these fields as follows:
   - **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:

- "Using the Mobile Devices Administration Center" on page 77
- "Creating a file subscription" on page 118
- "Specifying the encryption level"
- "Specifying the name of the source file" on page 120
- "Specifying subscription sets that use the subscription" on page 121
- "Flagging a subscription that references a changed file" on page 121

Specifying the encryption level

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.

Prerequisites:

Encryption support for Windows CE and Win32 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 Win32 platform clients and the DB2 Everyplace Sync Server. Encryption support on Palm OS requires Palm OS 3.3 or later.

Procedure:

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:

- "Using the Mobile Devices Administration Center" on page 77
- "Creating a file subscription" on page 118
- "Identifying the subscription" on page 119
- "Specifying the name of the source file"
- "Specifying subscription sets that use the subscription" on page 121
- "Flagging a subscription that references a changed file" on page 121

Specifying 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.

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.

Prerequisites:

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 Administrator.

Procedure:

Specify the name of the source file in the **Source filename** field on the Identification page of the Create File Subscription notebook.

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 121 for more information.

Because Palm OS mobile devices store the file name within the file itself, the **Target filename** field is ignored.

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Creating a file subscription" on page 118
- "Identifying the subscription" on page 119
- "Specifying the encryption level" on page 119
- "Specifying subscription sets that use the subscription"
- "Flagging a subscription that references a changed file"

Specifying subscription sets that use the subscription

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.

After you define the 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.

Procedure:

1. Go to the Subscription sets page of the Create File Subscription notebook.
2. 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, you will need to create a subscription set. You can click **Create** to open the Create Subscription set notebook 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 125 for more information on creating a subscription set.
3. 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.

You can also associate subscriptions with a subscription set using the Subscription set notebook. For more information, see "Specifying what information the subscription set accesses" on page 126.

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Creating a file subscription" on page 118
- "Identifying the subscription" on page 119
- "Specifying the encryption level" on page 119
- "Specifying the name of the source file" on page 120
- "Flagging a subscription that references a changed 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 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.

Procedure:

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:

- "Using the Mobile Devices Administration Center" on page 77
- "Creating a file subscription" on page 118
- "Identifying the subscription" on page 119
- "Specifying the encryption level" on page 119
- "Specifying the name of the source file" on page 120
- "Specifying subscription sets that use the subscription" on page 121

Creating a custom subscription

This section provides information about how to create a custom subscription using the Mobile Devices Administration Center. The topics covered are:

- "Creating a custom subscription"
- "Identifying the Custom subscription"
- "The default adapters" on page 123
- "Launch the customizer" on page 123
- "Assigning a subscription set to a Custom subscription" on page 124

Creating a custom subscription

Use the Create Custom Subscription notebook to create a custom subscription. A Custom subscription can be used to customize the default adapters or use a 3rd party adapter.

Procedure:

To create a custom subscription::

1. Open the Create Custom subscription notebook.
2. Identify the custom subscription.
3. Launch the customizer.
4. Assign subscription sets to the custom subscription.
5. Click **OK**.

Related concepts:

- "The default adapters" on page 123

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Creating an upload subscription" on page 112
- "Identifying the Custom subscription"
- "Launch the customizer" on page 123
- "Assigning a subscription set to a Custom subscription" on page 124

Identifying the Custom subscription

After authenticating a user, the 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 Sync Server must be able to uniquely identify each subscription by name.

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

Procedure:

1. Click the **Identification** tab of the Creating custom subscriptions notebook.
2. Fill in these fields:
   - **Name**

     Type a descriptive, unique 1- to 18-character name for the custom subscription
   - **Description**

     Optional: Type up to 128 characters of text that describes this subscription.
   - **Adapter**

     Click the down arrow and select the adpater that you want to use for your custom adapter.

Related concepts:

- "The default adapters"

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Creating an upload subscription" on page 112
- "Creating a custom subscription" on page 122
- "Launch the customizer"
- "Assigning a subscription set to a Custom subscription" on page 124

The default adapters

DB2 Everyplace Sync Server includes four default adapters:

- 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)

Custom adapters allow you to expand the capabilities of DB2 Everyplace Sync Server.

Related concepts:

- "The default adapters"

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Creating an upload subscription" on page 112
- "Creating a custom subscription" on page 122
- "Identifying the Custom subscription" on page 122
- "Launch the customizer"

Launch the customizer

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

Procedure:

On the Identification page, Click the **Launch Customizer** push button to launch the custom Java class file defined in the Customizer class name field of the custom adapter used by the subscription. The Customizer is only available if the adapter requires it. Otherwise, additional parameters for the custom subscription are entered in the **Other** field.

Related concepts:

- "The default adapters" on page 123

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Creating an upload subscription" on page 112
- "Creating a custom subscription" on page 122
- "Identifying the Custom subscription" on page 122
- "Assigning a subscription set to a Custom subscription"

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 Creating a custom subscription.

Procedure:

1. Go to the subscription sets page of the Create Custom Subscription notebook.
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 Custom Subscription notebook. The Create Subscription Set notebook 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:

- "The default adapters" on page 123

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Creating an upload subscription" on page 112
- "Creating a custom subscription" on page 122
- "Identifying the Custom subscription" on page 122
- "Launch the customizer" on page 123

Enabling groups to access their data

This section provides information about how to enable groups to access data using the Mobile Devices Administration Center. The topics covered are:

- "Enabling groups to access their data and files"
- "What is a subscription set?"
- "Identifying the subscription set" on page 126
- "Specifying what information the subscription set accesses" on page 126
- "Making the subscription set available to a group" on page 127

Enabling groups to access their data and files

To define a subscription set:

1. Open the Create Subscription set notebook. (See "Creating an object from the object tree" on page 70.)
2. Identify the subscription set.
3. Specifying what information the subscription set accesses.Evaluate your current groups of mobile users and determine what information they currently need to perform their work. Assign to the subscription set all DataPropagator or file subscriptions that represent this information.
4. Making the subscription set available to a groupEvaluate which groups require access to the subscriptions referenced in the subscription set. Associate these groups with the subscription set.
5. Click **OK**.

Related concepts:

- "What is a subscription set?"

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Identifying the subscription set" on page 126
- "Specifying what information the subscription set accesses" on page 126
- "Making the subscription set available to a group" on page 127

What is a subscription set?

An *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 "Registering a device" on page 88.

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Enabling groups to access their data and files"
- "Identifying the subscription set" on page 126

- “Specifying what information the subscription set accesses”
- “Making the subscription set available to a group” on page 127

Identifying the subscription set

After authenticating a user, the 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 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.

Procedure:

1. Click the **Identification** tab of the Creating upload subscriptions notebook.
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 concepts:

- “What is a subscription set?” on page 125

Related tasks:

- “Using the Mobile Devices Administration Center” on page 77
- “Enabling groups to access their data and files” on page 125
- “Specifying what information the subscription set accesses”
- “Making the subscription set available to a group” on page 127

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.

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.

For example, suppose 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 defining the subscriptions, associate the subscriptions with a subscription set and then assign that subscription set to the VNURSE group to enable the visiting nurses to 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.

Procedure:

To associate subscriptions with the subscription set:

1. Go to the Subscriptions page of the Create Subscription set notebook.
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.
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.

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, Create DataPropagator subscription notebook, Create File subscription notebook, or Create Custom subscription notebook now, or you can create a subscription later and associate a subscription set with it later.

Related concepts:

- "What is a subscription set?" on page 125

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Enabling groups to access their data and files" on page 125
- "Identifying the subscription set" on page 126
- "Making the subscription set available to a group"

Making the subscription set available to a group

To allow a group to access the DataPropagator and file 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 filesAdministrator.

Procedure:

1. Go to the Groups page of the Create Subscription set notebook.

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 will need to create a group. You can click **Create** to open the Create Group notebook now, or you can create a group later and associate a subscription set with it later. See "Creating a data synchronization group" on page 79 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 concepts:

- "What is a subscription set?" on page 125

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Enabling groups to access their data and files" on page 125
- "Identifying the subscription set" on page 126
- "Specifying what information the subscription set accesses" on page 126

Enabling and disabling synchronization

This section provides information about how to enable and disable synchronization using the Mobile Devices Administration Center. The topics covered are:

- "Enabling and disabling synchronization"
- "Enabling synchronization" on page 129
- "Disabling synchronization" on page 129

Enabling and disabling synchronization

Prerequisites:

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.

Procedure:

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. Provide a unique name for the user.
2. Assign the user to an enabled group.
3. Register the user's device.

You must create the user first in the Administrator before that user registers his or her device during the first synchronization. See "Registering a device" on page 88 for information on registering a device.

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Enabling synchronization"
- "Disabling synchronization"

Enabling synchronization

Procedure:

You can enable a group or user in one of the following ways:

- 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 Cernter.
  2. Right-click the object that you want to enable, then click **Edit**. The Edit notebook for the object opens.
  3. From any page of the Edit notebook, 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. See "Enabling and disabling synchronization" on page 128 for more information.
  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 Administrator.
  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. See "Enabling and disabling synchronization" on page 128 for more information.

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Enabling and disabling synchronization" on page 128
- "Disabling synchronization"

Disabling synchronization

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

Procedure:

Disable a group or user in one of the following ways:

- 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, then click **Edit**. The Edit notebook for the object opens.
  3. From any page of the Edit notebook, clear the **Enable synchronization** check box.
  4. 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**.

Related tasks:

- "Using the Mobile Devices Administration Center" on page 77
- "Enabling and disabling synchronization" on page 128
- "Enabling synchronization" on page 129

Chapter 6. The XML Scripting tool

This chapter provides information about how to use the XML Scripting tool. The topics covered are:

- "Utilizing the XML Scripting tool"
- "Overview of the XML Scripting tool"
- "Exporting XML scripts" on page 132
- "Invoking the XML Scripting tool from the command line" on page 133
- "Invoking the XML tool programmatically" on page 133
- "XML Scripting tool restrictions" on page 146

Utilizing the XML Scripting tool

The XML Scripting tool provides file based administration capabilities that enable automation and bulk processing of administrative tasks that are typically done through the Administrator. This section describes the tasks that you need to perform to utilize the XML Scripting tool. For more information see, "XML Scripting tool DTD" on page 205. The XML Scripting tool Java API is located in the `%DSYINSTDIR%`\doc\<lang>\javadoc

```
where <lang> is the country code
```

The following topics are discussed:

- Overview of the XML Scripting tool
- Using XML Scripting tool parameters
- XML Scripting tool example code

Related reference:

- "Invoking the XML Scripting tool from the command line" on page 133
- "Invoking the XML tool programmatically" on page 133
- "XML Scripting tool restrictions" on page 146
- "XML Scripting tool DTD" on page 205

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 mean 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 to 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 data 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:

- "Utilizing the XML Scripting tool" on page 131

Related tasks:

- "Exporting XML scripts"

Related reference:

- "Invoking the XML Scripting tool from the command line" on page 133
- "Invoking the XML tool programmatically" on page 133
- "XML Scripting tool restrictions" on page 146
- "XML Scripting tool DTD" on page 205

Exporting XML scripts

The following procedure shows the typical steps that an administrator or application developer will do to export XML scripts.

Procedure:

1. Export the information from the control database into am 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" on page 133 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 Administrator, you must perform it before you export the Mobile Devices Administrator 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 205.

Related concepts:

- "Utilizing the XML Scripting tool" on page 131
- "Overview of the XML Scripting tool" on page 131

Related reference:

- "Invoking the XML Scripting tool from the command line" on page 133
- "Invoking the XML tool programmatically" on page 133
- "XML Scripting tool restrictions" on page 146

- “XML Scripting tool DTD” on page 205

Invoking the XML Scripting tool from the command line

Table 6 shows the command line parameters that you can use with the XML Scripting tool. For example,

```
dsyadminxml -d <mynewsettings.xml>

dsyadmnxml -x <mycurrentsettings.xml>
```

Table 6. XML Scripting tool parameters

Parameter	Function
– x	Exports data from the control database into XML
– d	Applies the XML in your XML script file to the control database
– v	Verifies the XML in your XML script file
– ?	Displays help
– x – add	Generates XML that can be used to add the synchronization objects. This is the default behavior of -x.
– x – remove	Generates XML that can be used to remove the synchronization objects.

Related concepts:

- “Utilizing the XML Scripting tool” on page 131
- “Overview of the XML Scripting tool” on page 131

Related tasks:

- “Exporting XML scripts” on page 132

Related reference:

- “Invoking the XML tool programmatically”
- “XML Scripting tool restrictions” on page 146
- “XML Scripting tool DTD” on page 205

Invoking the XML tool programmatically

The following code example illustrates 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 may 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 programmar 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:

- "Utilizing the XML Scripting tool" on page 131
- "Overview of the XML Scripting tool" on page 131

Related tasks:

- "Exporting XML scripts" on page 132

Related reference:

- "Invoking the XML Scripting tool from the command line" on page 133
- "XML Scripting tool DTD" on page 205

XML sample scripts

This topic provides example for the following:

- "JDBC subscriptions"
- "DataPropagator subscriptions" on page 141
- "Creating a Group or a User" on page 144
- "Subscription sets" on page 145

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 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 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 may specify indexes for the table columns using <AddIndex> tag.
Index names need to be unique for the whole system for the sake of simplicity.

Below 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 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>VNPERSON</TargetTable>
         <SourceSchema>DSYSAMPLE</SourceSchema>
         <SourceTable>VNPERSON</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 may 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 may 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>VNPERSON</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 are 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 (will not be syncable) 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>
          <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>
   </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. Since 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 may 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>VNPERSON</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. Since 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>VNPERSON</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 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
Sync Server never existed. XML Tool picks up from the point where
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
Sync Server via 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 may specify indexes for the table columns using
<AddIndex> tag. Index names need to be unique for the whole system for
the sake of simplicity.

Below 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>
      <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>
   <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

```
<AddGroup> tag defines a Group for Sync Server. Users that can
synchronize data with 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 <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 may 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'.

At last, 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 it's 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>
```

Subscription sets

<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 JDBCSUB_2 and FILESUB_3 where this example assumes that JDBCSUB_2 and FILESUB_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, JDBCSUB_2 will be synchronized before FILESUB_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>JDBCSUB_2</Name>
      </IncludeSubscription>
      <IncludeSubscription>
       <Name>FILESUB_3</Name>
      </IncludeSubscription>
   </AddSubscriptionSet>
</Root>
```

```
Example of changing an existing subscription set:

This example adds a subscription named 'JDBCSUB_4'
(assuming JDBCSUB_4 already exists)
into the subscription set 'SUBSET_2' and makes this new susbcription
the 1st 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>JDBCSUB_4</Name>
        <Order>1</Order>
       </IncludeSubscription>
      </SubscriptionSet>
   </AlterSubscriptionSet>
</Root>
```

Related concepts:

- "Utilizing the XML Scripting tool" on page 131
- "Overview of the XML Scripting tool" on page 131

Related tasks:

- "Exporting XML scripts" on page 132

Related reference:

- "Invoking the XML Scripting tool from the command line" on page 133
- "XML Scripting tool DTD" on page 205

XML Scripting tool restrictions

The following restrictions apply to the XML Scripting tool:

- To create DataPropagator subscriptions for the Sync Server, DataPropagator replication must be set up between the source and the mirror using the DataPropagator tools.
- When creating a Data Propagator 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 Data Propagator subscription.

Related concepts:

- "Utilizing the XML Scripting tool" on page 131
- "Overview of the XML Scripting tool" on page 131

Related tasks:

- "Exporting XML scripts" on page 132

Related reference:

- "Invoking the XML Scripting tool from the command line" on page 133
- "Invoking the XML tool programmatically" on page 133
- "XML Scripting tool DTD" on page 205

Chapter 7. Monitoring client synchronization status

This chapter provides information about how to monitor client synchronization status using the Synchronization Monitor tool. The topics covered are:

Monitoring client synchronization status

You can monitor the synchronization status of a single user or check the status of all users who are synchronizing on your system. DB2 Everyplace Sync Server provides a graphical interface called the Synchronization Monitor that shows you the synchronization status of the users on your system. This section describes:

- Accessing the Synchronization Monitor tool
- Using the Synchronization Monitor tool
  1. Setting an automatic refresh
  2. Selecting a view
  3. Customizing the column display
  4. Exporting data
- "Accessing the Synchronization Monitor tool"
- "Using the Synchronization Monitor tool" on page 148
- "Setting an automatic refresh" on page 148
- "Selecting a view" on page 149
- "Customizing the column display" on page 149
- "Exporting data" on page 149

Accessing the Synchronization Monitor tool

Procedure:

1. To open the Synchronization Monitor tool: Set up the Control Database. DB2 Everyplace populates the Sync Monitor interface with synchronization information from the log table which is located in the control database. To use the Sync Monitor interface, you must type the name of the control database that you are using for the Sync Server. To set the control database:
   a. Select **Monitor –> Set Control Database**. The Set Control Database window opens.
   b. Type the name of the control database in the **Control database name** field.
   c. Click **OK**.
2. Locate the `/Server/bin` directory.
3. Double click the `dsysyncmonitor.bat` file. The Sync Status of Users window opens.

If synchronization is ***not*** occurring, the Sync Status of Users window will be empty.

Related tasks:

- “Monitoring client synchronization status” on page 147
- “Using the Synchronization Monitor tool”
- “Setting an automatic refresh”
- “Selecting a view” on page 149
- “Customizing the column display” on page 149
- “Exporting data” on page 149

Using the Synchronization Monitor tool

The Synchronization Monitor tool allows system administrators to view information about the synchronization status of users. For more information, the Synchronization Monitor tool API is located in `%DSYINSTDIR%`\doc\<lang>\javadoc `where <lang> is the country code`

The tasks that you can performed from the Sync Status of Users window include:

- Setting an automatic refresh
- Selecting a view
- Customizing the column display
- Exporting data

Related tasks:

- “Monitoring client synchronization status” on page 147
- “Accessing the Synchronization Monitor tool” on page 147
- “Setting an automatic refresh”
- “Selecting a view” on page 149
- “Customizing the column display” on page 149
- “Exporting data” on page 149

Setting an automatic refresh

You can set the refresh rate to automatically update the synchronization information in the Sync Monitor GUI.

Procedure:

1. Click the down arrow in the **Seconds** field.
2. Type or select the number of seconds that you want to use for the refresh rate.
3. Click **Refresh**.

Related tasks:

- “Monitoring client synchronization status” on page 147
- “Accessing the Synchronization Monitor tool” on page 147
- “Using the Synchronization Monitor tool”
- “Selecting a view” on page 149
- “Customizing the column display” on page 149
- “Exporting data” on page 149

Selecting a view

You can customize the view in the Sync Status window based on synchronization status of users. To customize the display, click the down arrow in the **Show** field and select the status that you want to view.

Related tasks:

- "Monitoring client synchronization status" on page 147
- "Accessing the Synchronization Monitor tool" on page 147
- "Using the Synchronization Monitor tool" on page 148
- "Setting an automatic refresh" on page 148
- "Customizing the column display"
- "Exporting data"

Customizing the column display

The view in the Sync Status window can be customized to display the columns in the order of your preference.

Procedure:

1. Select **View –> Customize Columns**. The Customize Columns window opens.
2. Select the check box next to the column that you would like to move.
3. Click **Move Up** or **Move Down** until the selected column is in the desired position.
4. Click **OK**.

Related tasks:

- "Monitoring client synchronization status" on page 147
- "Accessing the Synchronization Monitor tool" on page 147
- "Using the Synchronization Monitor tool" on page 148
- "Setting an automatic refresh" on page 148
- "Selecting a view"
- "Exporting data"

Exporting data

You can export the data from the Sync Status window to a text file or copy it into a spread sheet application.

Procedure:

1. To export the data to a text file: Click and Highlight the rows that you want to export.
2. Select **Edit –> Export**. The File Save window opens.
3. Type or select the directory where you want the text file to be saved.
4. In the **Name** field, type the name of the text file.
5. Click **OK**.

Related tasks:

- "Monitoring client synchronization status" on page 147
- "Accessing the Synchronization Monitor tool" on page 147

- “Using the Synchronization Monitor tool” on page 148
- “Setting an automatic refresh” on page 148
- “Selecting a view” on page 149
- “Customizing the column display” on page 149

Chapter 8. Synchronization problems

This chapter provides information about how to troubleshoot synchronization problems. The topics covered are:

- “Handling synchronization problems”
- “The order of synchronization and reception of error messages”
- “Viewing the error log to diagnose problems” on page 152
- “Purging error log entries automatically” on page 153
- “Defining the tracing level” on page 153
- “Providing your own error-handling logic” on page 154
- “Viewing the log on the client” on page 156
- “Resetting the user ID” on page 157

Handling synchronization problems

When synchronization is interrupted, the Sync Server writes messages to the log in the administration control database. The following topics explain how to:

- Understand the order of synchronization and reception of error messages
- View the log through the Administrator to determine the cause of the problem
- Purging error log entries automatically
- Defining the tracing level
- Add error-handling logic to automatically perform actions in response to an error message
- View the log on the client device
- Reset the user ID

Related concepts:

- “The order of synchronization and reception of error messages”

Related tasks:

- “Viewing the error log to diagnose problems” on page 152
- “Purging error log entries automatically” on page 153
- “Defining the tracing level” on page 153
- “Providing your own error-handling logic” on page 154
- “Viewing the log on the client” on page 156
- “Resetting the user ID” on page 157

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 Administrator. 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 Administrator . The subscriptions listed in the Editing Subscription Set notebook or the tables displayed in the Editing Subscription notebook 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 Sets, Create Subscription Sets, Change Group, Create Group, and Define Replication Subscriptions notebooks.

If a record was rejected by the 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 synchronization problems" on page 151

Related tasks:

- "Viewing the error log to diagnose problems"
- "Purging error log entries automatically" on page 153
- "Defining the tracing level" on page 153
- "Providing your own error-handling logic" on page 154
- "Viewing the log on the client" on page 156
- "Resetting the user ID" on page 157

Viewing the error log to diagnose problems

When you encounter synchronization problems, you can use the Administrator to view the error log for troubleshooting purposes.

Procedure:

1. To access the log, 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 Sync Server reporting this log entry.

To determine the actions that you should take for a particular message, see "Error messages" on page 161, which lists messages by code along with an explanation and suggested actions that you can take to resolve the problem.

Additional logs are created in the form of trace files. Trace files are stored, by default, in the `\SyncServer\Server\logs` directory. This location is defined by the `Trace.Path` value found in the `DSYGdflt.properties` file. They have a `.trace` suffix. You can use a text editor to view the trace files.

Related concepts:

- "Handling synchronization problems" on page 151
- "The order of synchronization and reception of error messages" on page 151

Related tasks:

- "Purging error log entries automatically"
- "Defining the tracing level"
- "Providing your own error-handling logic" on page 154
- "Viewing the log on the client" on page 156
- "Resetting the user ID" on page 157

Purging error log entries automatically

You can specify how many days the entries in the error log in the Administrator 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.

Procedure:

1. To specify the days that the entries should be kept, Use the `dsysetproperty` tool to set the `DSYGdflt Log.KeepDays` property. You can also set a maximum number of log entries which will be stored before pruning will take place. This is dictated by the `DSYGdflt Log.PruneToSize` property, which defaults to 10000 entries.

Related concepts:

- "Handling synchronization problems" on page 151
- "The order of synchronization and reception of error messages" on page 151

Related tasks:

- "Viewing the error log to diagnose problems" on page 152
- "Defining the tracing level"
- "Providing your own error-handling logic" on page 154
- "Viewing the log on the client" on page 156
- "Resetting the user ID" on page 157

Defining the tracing level

By default, the Sync Server only logs error messages in the trace files. However, for diagnostic purposes, you may want to turn on tracing to include more detailed information. Use **DSYTrace** to turn on or off tracing.

Procedure:

To turn on tracing:

1. Open a command prompt.
2. Change to the `%DSYINSDIR%/Server/bin` directory.

3. At the command prompt, type:
   ```
   DSYTrace 1 -console
   ```

All trace messages are now logged to a .trace file.

To turn off tracing:

1. Open a command prompt.
2. Change to the %DSYINSDIR%/Server/bin directory.
3. At the command prompt, type:
   ```
   DSYTrace 0 -console
   ```

Tracing is now turned off. Only error messages are now logged to a .trace file.

The DSYTrace command changes the Trace.Level in file DSYGdflt.properties in directory \%DSYINSTDIR%\Server\properties\com\ibm\mobileservices\. 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.

Related concepts:

- "Handling synchronization problems" on page 151
- "The order of synchronization and reception of error messages" on page 151

Related tasks:

- "Viewing the error log to diagnose problems" on page 152
- "Purging error log entries automatically" on page 153
- "Providing your own error-handling logic"
- "Viewing the log on the client" on page 156
- "Resetting the user ID" on page 157

Providing your own error-handling logic

Frequently, error messages 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.

Procedure:

1. To define automatic processing of errors: Using a text editor of your choice, open the following file for editing:
   ```
   DSYUserExits.properties
   ```

   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 messages section 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.
```

2. 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 161 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.

   The action to be performed must be a reference to a valid routine or program.

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 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, ...
```

Related concepts:

- “Handling synchronization problems” on page 151
- “The order of synchronization and reception of error messages” on page 151

Related tasks:

- “Viewing the error log to diagnose problems” on page 152
- “Purging error log entries automatically” on page 153
- “Defining the tracing level” on page 153
- “Viewing the log on the client”
- “Resetting the user ID” on page 157

Viewing the log on the client

On a Palm device or emulator, if you are using the IBM Sync, it keeps a synchronization log (`logdb-isyn`). To view the log click the **Log** button.

The contents of the synchronization log are overwritten every time when a new synchronization starts.

On a Palm OS device or emulator, the filename of the log is LOGDB-ISYN.pdb. If you want to save its contents for debugging purposes, you can use any utility that can view the contents of a .pdb file to open this file. In addition. to the synchronization log that the IBM Sync provides, the synchronization engine generates a trace file (trace-isyn), which is required when customer services are needed.

Related concepts:

- "Handling synchronization problems" on page 151
- "The order of synchronization and reception of error messages" on page 151

Related tasks:

- "Viewing the error log to diagnose problems" on page 152
- "Purging error log entries automatically" on page 153
- "Defining the tracing level" on page 153
- "Providing your own error-handling logic" on page 154
- "Resetting the user ID"

Resetting the user ID

Occasionally, you might find it necessary to reset the user ID for a mobile device. For example, you might reset the ID for a mobile device that is having a technical problem. If the user ID is being reset, be sure that you do not attempt to synchronize until the reset is complete. During the reset process all configuration data used for synchronization is cleaned up. No manual clean up is required.

Caution: Group, user, and device resets will cause changed data on individual devices to be lost.

To use the Mobile Device Administration Center to reset the user ID for a mobile device, click **User —> Reset**.

To reset a user ID without using Mobile Device Administration Center, run the *dysreset.bat* tool from a command line, using 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'.
```

Related concepts:

- "Handling synchronization problems" on page 151
- "The order of synchronization and reception of error messages" on page 151

Related tasks:

- "Viewing the error log to diagnose problems" on page 152
- "Purging error log entries automatically" on page 153
- "Defining the tracing level" on page 153
- "Providing your own error-handling logic" on page 154

- “Viewing the log on the client” on page 156

Part 4. Reference

Chapter 9. Sync Server reference information

This chapter provides Sync Server reference information. The topics covered are:

- "Error messages"
- "Command line scripts for the DB2 Everyplace Sync Server" on page 197
- "Syntax for filters" on page 202
- "XML Scripting tool DTD" on page 205
- "Data type mapping and restrictions" on page 220

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 appendix with "Handling synchronization problems" on page 151 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 reset registered user of the device and try again.

DSYC2000C **Requested data changes rejected.**

Explanation: Some of the data changes that the Sync Client requested have been rejected due to data conflicts or no permission.

User Response: Check with the Sync Server administrator to ensure that the user has permission to perform the desired operation. Since the requested changes have been rejected, the user has to reapply the changes, or have the Sync Server administrator reset the user and try again. If problems persist, contact IBM software support.

DSYC300E **Failed to open adapter:** *adapter name*.

Explanation: The Sync Client could not open the adapter library after use.

User Response: Check that the library 'adapter name' is present on the device. If the library name is unknown, check with the Sync Server administrator. Otherwise, have the Sync Server administrator reset the user and try again.

DSYC301E **Failed to load adapter:** *adapter name*.

Explanation: The Sync Client could not load the adapter library after use.

User Response: Check that the library for 'adapter name' is present on the device. If the library name is unknown, check with the Sync Server administrator. Otherwise, have the Sync Server administrator reset the user and try again.

DSYC302E **Failed to close adapter:** *adapter name*.

Explanation: The Sync Client could not close the adapter library after use.

User Response: Check that the library for 'adapter name' is present on the device. If the library name is unknown, check with the Sync Server administrator. Otherwise, have the Sync Server administrator reset the user and try again.

DSYC303E Configuration synchronization failed, synchronization aborted.

Explanation: The Sync Client did not properly synchronize the configuration information.

User Response: Have the Sync Server Administrator reset the user, and try again. If problems persist, contact IBM software support.

DSYC304E Authentication failed - synchronization aborted.

Explanation: The provided username/password does not pass the authentication on the Sync Server.

User Response: Make sure that the password is entered correctly, and try again. If problems persist, contact the Sync Server administrator.

DSYC306E Authentication failed (invalid encryption key) - synchronization aborted.

Explanation: The Sync Server could not successfully recognize and encrypt/decrypt the message from the client.

User Response: Make sure that encryption is supported on the Sync Client platform, do a soft reset on the device and try again. If problems persist, contact the Sync Server administrator.

DSYC307E Client encryption/decryption failed - synchronization aborted.

Explanation: The Sync Client could not successfully encrypt/decrypt the received message.

User Response: Make sure that encryption is supported on the client's operating system, do a soft reset on the device and try again. If problems persist, contact IBM software support.

DSYC308E Encryption not available.

Explanation: Encryption is not supported for Palm OS 3.2 and earlier. The encryption library is not installed or the path is not set correctly.

User Response: Make sure that the client's operating system supports the encryption, the encryption library is properly installed, and the path is set correctly. If problems persist, contact the Sync Server administrator.

DSYC309E Failed to open encryption library.

Explanation: Encryption is not supported for Palm OS 3.2 and earlier. The path for encryption is not set correctly.

User Response: Make sure that the operating system supports the encryption, the path for encryption is set correctly. If problems persist, contact the Sync Server administrator.

DSYC310E Incompatible Sync Client version.

Explanation: The version for the Sync Client is not compatible with the Sync Server.

User Response: Upgrade the Sync Client to the latest version, and try again. If problems persist, contact the Sync Server administrator.

DSYC311E Subscription not found.

Explanation: The Sync Server cannot find the subscription that the client is requesting. It may be because the configuration has changed since the client starts synchronization.

User Response: Try to sync again, or have the Sync Server administrator reset the user and sync again. If problems persist, contact the Sync Server administrator.

DSYC312E Subscription blocked by the Server.

Explanation: The subscription has been blocked by the Sync Server because the Sync Server administrator is performing maintenance on the control database or modifying the configuration.

User Response: Please wait and try to sync again. If problems persist, contact the Sync Server administrator.

DSYC313E Invalid synchronization session.

Explanation: The synchronization session of the client and the one persisting on the Sync Server do not match. The client configuration may be corrupted or overwritten.

User Response: Have the Sync Server administrator reset the user. Try to sync again. If problems persist, contact the Sync Server administrator.

DSYC314E The user does not belong to any Sync Group.

Explanation: The user has not been added to any group with synchronization privilege.

User Response: Check that the username is entered correctly and have the Sync Server administrator verify that the user has been added to a synchronization-enabled group. Try to sync again. If problems persist, contact the Sync Server administrator.

DSYC315E Failed to register the device for the user.

Explanation: The Sync Server cannot register the device. The Sync Client has changed its user information, and therefore, no longer matches the registered user information on the Sync Server.

User Response: Check that the username is entered correctly and have the Sync Server administrator reset the user. Try to sync again. If problems persist, contact the Sync Server administrator.

DSYC316E Subscription definition altered.

Explanation: The definition of a subscription has changed since the last time the client was configured. The Sync Server administrator is modifying the configuration.

User Response: Wait for a while and try to sync again. If problems persist, contact the Sync Server administrator.

DSYC400E Failed to allocate adapter resources.

Explanation: The Sync Client could not allocate adapter-specific resources (for example, database connections, or statement handles) for synchronizing the subscribed data.

User Response: Check that the required adapter-specific resources are not occupied and try again. If problems persist, consult the DB2 Everyplace Application Development Guide or contact IBM software support.

DSYC401E Failed to connect to the target data.

Explanation: The 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 Sync Server administrator reset the user and try again. If problems persist, contact IBM software support.

DSYC402E Failed to disconnect from the target data.

Explanation: The 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 Sync Server administrator reset the user and try again. If problems persist, contact IBM software support.

DSYC403E Missing subscription information from server.

Explanation: The Sync Client received unexpected empty or missing subscription information (for example, no subscribed table information) from the server.

User Response: There may be missing information in the subscription. Contact the Sync Server administrator.

DSYC412E Unexpected message format.

Explanation: The Sync Client does not recognize some message information from the server during synchronization of a subscription.

User Response: Have the Sync Server administrator reset the user and try again. If problems persist, contact IBM software support.

DSYC413E Requested data not found.

Explanation: The Sync Client cannot find the target data (for example, table not found) for synchronization.

User Response: The target data may have been deleted by another application. Have the Sync Server administrator reset the user and try again. If problems persist, contact IBM software support.

DSYC414E Unexpected end of data.

Explanation: The Sync Client has encountered unexpected missing or end-of-data during synchronization of a subscription.

User Response: Have the Sync Server administrator reset the user and try again. If problems persist, contact IBM software support.

DSYC415E Data too long for corresponding field.

Explanation: The Sync Client received data that was too large (for example, over-sized file). Data may have been truncated.

User Response: The data from the Sync Server may be invalid. Check with the Sync Server administrator.

DSYC417E Server reported user or device not enabled.

Explanation: The Sync Server administrator has not enabled the user to synchronize.

User Response: Make sure that the username is entered correctly, and have the Sync Server administrator enable the user, and try again. If problems persist, contact the Sync Server administrator.

DSYC418E Server reported unknown exceptions.

Explanation: The Sync Server could not synchronize this user due to an unexpected error.

User Response: Contact the Sync Server administrator.

DSYC420E Attempt to update read-only data.

Explanation: The subscription adapter attempts to modify the target data which is read-only.

User Response: Check that the user has write capability on the target data, and try to sync again. If problems persist, contact the Sync Server administrator.

DSYC421E Illegal operation on the data.

Explanation: The subscription adapter detected problems performing operations on the target data.

User Response: Check with the administrator and make sure the user has permission to perform the operations. Have the Sync Server administrator reset the user and try again. If problems persist, 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 are entered correctly, and that the user is authorized to access the requested data. Have the Sync Server administrator reset the user and try again. If problems persist, contact the 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 Sync Client has permission to modify the target data, and that it is not being used by another application. Have the Sync Server administrator reset the user and try again. If problems persist, 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 columns types which are not supported by DB2 Everyplace.

User Response: Have the Sync Server administrator verify that all the subscribed data are supported (for example, file type and column types), and try to sync again. If problems persist, 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 Sync Client has a problem opening a connection with the 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 Sync Server is currently operational. Click on ″Retry″. If problems persist, contact IBM software support.

DSYC601E Failed to establish connection.

Explanation: The Sync Client has a problem opening connection with the 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 Sync Client has access to the Sync Server through either a serial or USB cable, a modem, or a network connection. 3) The 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 problems persist, contact IBM software support.

DSYC602E Failed to send request.

Explanation: The Sync Client has successfully connected to the Sync Server, but has a problem sending a request to the server. This could be due to the following: 1) Server is not up 2) Network connection is lost.

User Response: Check to ensure: 1) The Sync Client has access to the Sync Server through either a serial or USB cable, a modem, or a network connection. 2) The 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 problems persist, contact IBM software support.

DSYC603E Failed to receive reply.

Explanation: The Sync Client has successfully sent a request to the Sync Server, but has a problem receiving a reply from the server, or the received message is corrupted or in an unexpected format. This could be due to the following: 1) Server is not up 2) Network connection is lost.

User Response: Check to ensure: 1) The 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 problems persist, contact IBM software support.

DSYC604E Timeout while receiving reply.

Explanation: The Sync Client has successfully connected sent a request to the Sync Server, but has not received a reply from the server before timeout. 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 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 problems persist, contact IBM software support.

DSYC605E Failed to receive acknowledgement.

Explanation: The Sync Client has successfully sent a request and received a reply from the Sync Server, but does has not 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 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 problems persist, contact IBM software support.

DSYC606E Failed to open the Net library.

Explanation: The 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 problems persist, contact IBM software support.

DSYC608E Failed to close the Net library.

Explanation: The Sync Client has a problem closing the Network library.

User Response: Try a soft reset on the device, and synchronize again. If problems persist, contact IBM software support.

DSYC609E Failed to resolve hostname.

Explanation: The 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 Sync Client could not allocate sufficient memory for completing 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 problems persist, 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: Check that the user is authorized to synchronize to the Sync Server. Try to synchronize again. If problems persist, contact the Sync Server administrator.

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 Sync Server is not installed properly.

User Response: Check to ensure: 1) Server URL (Uniform Resource Locator) is correctly entered; 2) The Sync Server is currently operational; 3) The Sync Server is installed properly. Try to sync again. If problems persist, contact the Sync Server administrator.

DSYC613E Sync Server Error.

Explanation: An internal error has occurred on the Sync Server.

User Response: Restart the Sync Server and try to synchronize again. If problems persist, contact IBM software support.

DSYC614E Server not responding.

Explanation: Servers are not responding to the client requests. All servers are not available (either busy or blocked).

User Response: Check with the Sync Server administrator that the Sync Server is available. Try to synchronize again. If problems persist, 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 Sync Server and Sync Client have been configured for SSL (Secure Socket Layer).

DSYC698E Internal error: invalid or inconsistent state.

Explanation: The subscription adapter encountered errors or inconsistent state from the DB2 Everyplace database engine when synchronizing the subscribed data.

User Response: Check the Sync Client configuration log, and identify the subscription type (config, table, or file). If applicable, consult the DB2 Everyplace Application Development Guide with the provided SQL state in the error message. Have the Sync Server administrator reset the user and try again. If problems persist, contact IBM software support.

DSYC699E Unknown network error.

Explanation: The Sync Client has encountered an unknown error while communicating with the 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 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 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 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 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 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 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 UDB Replication Guide and Reference for more information on 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 UDB 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 UDB 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 A request to initiate replication on database *mirror database name* **was successfully communicated.**

Explanation: A request to initiate replication on the mirror database was successfully communicated.

User Response: No action is required.

DSYD032E Look-up table *table name* **is not found in the mirror database, but is mentioned in the user filter of** *mirror table name* **in** *subscription name***.**

Explanation: The filter makes reference to a table that does not exist in the mirror database.

User Response: Correct the filter in the Mobile Device Administration Center to make it refer to only those tables that are replicated to the mirror database by deleting the subscription and creating a new one with the correct filter.

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 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 Default value of *column name* **is undefined in group** *group name* **or user** *user name* **for table** *table name* **in database** *database name***.**

Explanation: A required default value is undefined for an unsubscribed column.

User Response: Ensure that the parameters of vertical filtering are defined and spelled correctly. Make corrections in the Subscription, Group, and/or User folders in the Mobile Device Administration Center.

DSYD040E The JDBC replication infrastructure is incompatible with this installation of DB2 Everyplace Sync Server.

Explanation: The infrastructure that supports the replication of JDBC subscriptions is not at the level required by this installation of Sync Server.

User Response: Run the migration utility that came with this installation of Sync Server. The utility migrates JDBC subscriptions from previous versions of Sync Server to the version of Sync Server installed.

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 sync server installation: *level*

Explanation: This 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 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 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 Sync Server 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 since 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.

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 may include additional information that may 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 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 may 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 may 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 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 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 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 DataPropragator 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. Once 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 may 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 may 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 Sync Server servers were unable to block subscription *subscription name*. **Action canceled.**

Explanation: One or more 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 may be restarted and/or the action may be attempted again.

DSYM158E Unable to connect to the LDAP server *WEBSERVICE_SOAP_ROUTER* **using login** *user name*.**\n\nSpecify a valid user name and/or password and try again or modify the DSYLDAP.properties to change the LDAP default connection information.\n\nLDAP 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/mobileserivces/DSYLDAP.properties. Specify a valid WEBSERIVCE_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 matching DB2e* were found in the LDAP server *WEBSERVICE_SOAP_ROUTER*.

Explanation: No groups matching DB2e* were found at the specified LDAP server.

User Response: Define at least one DB2* 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 may not exceed 255 characters. DB2 Everyplace requires that the total length of all primary key columns may not exceed 1024 characters.

User Response: Select a valid table and try again.

DSYM165E One or more Sync Server servers were not offline. All sync servers must be marked as offline to create a DataPropagator subscription. Action canceled.

Explanation: One or more Sync Server servers were not offline. All 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 may be viewed, but any changes made will be ignored because one or more Sync Server servers were not offline. All sync servers must be marked as offline to edit a DataPropagator subscription.**

Explanation: One or more Sync Server servers were not offline. All 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 may be restarted and/or the action may 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 may be restarted and/or the action may 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 may be restarted and/or the action may 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.

DSYM178I DB2 Everyplace Mobile Device Administration Center has detected that DB2 UDB v8.1 is installed. You are now running the UDB v7.2 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 v8.1 is installed. Since DB2 Everyplace Sync Server does not fully support UDB v8.1 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 UDB 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 may 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 data types 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 may be a maximum of 18 characters in length.

User Response: Specify a valid target table name and try again.

DSYS000I **Complete synchronization request received;** *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 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 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 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 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* LDAP group. Session aborted.

Explanation: The specified user is a member of more than one group named DB2e* which has subscriptions assigned to it. An LDAP may 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.

DSYT001E User *user name* cannot override data filter *data filter name* because the data filter is not defined for this users 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 users 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 may be caused by invalid column references, table name or index may 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 may be marked as running, thus admin tool is not able to communicate with it, or a running server may not be responding. There may 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 may 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 may 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 Sync Server. Replication process may not be able to figure out how to replicate this column. Setup process will continue, however, Sync Server may 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 may be marked as running, thus admin tool is not able to communicate with it, or a running server may not be responding, or the mirror database may be down. There may 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 Sync Server.

Explanation: XML Tool failed to create a user.

User Response: This maybe be because the user already exists, or one of it's properties were invalid. There may be more information displayed as the root cause.

DSYT031E There was an error creating an LDAP group and associating it with Sync Server.

Explanation: XML Tool failed to create a group.

User Response: This maybe be because the group already exists, or one of it's properties were invalid. There may 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 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 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'.**

Explanation: In LDAP enabled Sync Server installations, to be able to synchronize, a user needs to belong to 1 'DB2e' group whose name starts with 'DB2e'. This 'DB2e' group is used by Sync Server for synchronization purposes.

User Response: Provide a valid group name that starts with 'DB2e' in addition to other groups this LDAP user may already belong to.

Related concepts:

- “Handling synchronization problems” on page 151

Command line scripts for the DB2 Everyplace Sync Server

The following tables list the scripts that are available to operate and administer the DB2 Everyplace Sync Server, and the proper usage to run these scripts. There are two main locations for the scripts.

DB2 Everyplace Sync Server and Mobile Device Administration Center commands location:

- On Windows, these scripts are located in `%DSYINSTDIR%/Server/bin`, where %DSYINSTDIR% is the directory where DB2 Everyplace is installed.
- On UNIX and Linux platforms, these scripts are located in the `$HOME/db2everyplace81/Server/bin` directory, where $HOME is the home directory of the DB2 Everyplace Sync Server instance. You must run these commands from the $HOME/db2everyplace81/Server/bin directory and the command must be preceded with "./". For example: `./<command name>`.

  Notes:

  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-install configuration tasks specific to the operating system of your server prior to using the DB2 Everyplace Sync Server and Mobile Devices Administration Center. For example, you must export the DISPLAY variable to a valid display prior to executing these scripts. For more information, refer to the DB2 Everyplace Install Notes.

DB2 Everyplace Sync Server instance administration commands location:

- For AIX:

  `/usr/lpp/DB2Everyplace81/instance`
- For Linux (x86, iSeries):

  `/opt/DB2Everyplace81/instance`
- For Solaris:

  `/opt/DB2Everyplace81/instance`
- For Windows platforms, there are no scripts for managing instances.

Scripts used for installing DB2 Everyplace into WebSphere are documented in the section called "The WAS installation scripts" in the *DB2 Everyplace Installation and User’s Guide*.

The commands for UNIX/Linux and Windows platforms are very similar, except that UNIX and Linux scripts have a .sh extension, and Windows batch files have a .bat extension. The following tables assume Windows usage, but the command and syntax are functionally equivalent on UNIX/Linux platforms.

Table 7. Commands, for UNIX, Linux, and Windows

Command	Purpose	Usage

`dsyadmin.bat`	This command starts the DB2 Everyplace Mobile Device Administration Center (MDAC). MDAC provides a GUI to administer users, groups, subscription sets, and subscriptions.
	See "What is the Mobile Devices Administration Center?" on page 64 for more information about using MDAC.
	For Linux servers, the DB2 JDBC Applet Server must be started prior to starting MDAC. 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.sh.
		For Windows:
		`dsyadmin.bat`
		For Linux:
		`db2jstrt` *`[a free port number]`*
		`./dsyadmin.sh` *`[DB2 JDBC Applet Server port]`*
		For Solaris and AIX:
		`./dsyadmin.sh`

`dsyadminxml.bat`	This command runs the MDAC XML to manage MDAC configurations from the command line using XML files. The XML tool can perform add, remove, and alter operations on MDAC objects. MDAC objects include DB2 Everyplace users, groups, subscription sets and subscriptions.
	See "Utilizing the XML Scripting tool" on page 131 for complete usage information.
	For help, type
	`dsyadminxml.bat`
	with no arguments:
		Command line parameters:
		`-x|-d filename`
		`-x filename [-add] [-remove]`
		• `-x` create XML file from control database
		• `-d` apply XML file to control database
		• `filename` XML file name to process. -x will override file if it already exists.
		• `[-add] [-remove]` are XML Types that add to the sync server control database or remove what is already there

`dsyblocksubscription.bat`	This command blocks the subscription name provided in the command argument. More than one subscription can be named at once on the command line. Blocking a subscription will instruct the Sync Server to reject new changes from being applied to the subscription data.	`dsyblocksubscription.bat` *`[subscription]`*`+`
		`[subscription]` is the subscription to block on the server
		For example:
		`dsyblocksubscription subscriptionName1`
		would block the subscription with the name 'subscriptionName1'.

`dsyctldb.bat`	This command creates the DB2 Everyplace Control database. This command is normally run automatically by the DB2 Everyplace installation.	`dsyctldb.bat` *`<db2instance> <db2instance password>`*

Table 7. Commands, for UNIX, Linux, and Windows (continued)

Command	**Purpose**	**Usage**

`dsyjdbcmigration.bat`	This command migrates JDBC subscriptions from Sync Server v8.1.2 and earlier to Sync Server v8.1.4 and later.	`dsyjdbcmigration.bat [sourceDatabaseUrl sourceTableSpace] | [sourceDatabaseUrl sourceTableSpace sourceTableName]`
		where
		• `sourceDatabaseUrl` is the URL of the source database to migrate
		• `sourceTableSpace` is the name of the tablespace in which Sync Server control tables should be created
		• `sourceTableName` is the name of the source table to migrate
		For example, to migrate all subscriptions, type:
		`dsyjdbcmigration.bat`
		To migrate the subscriptions on the VNURSE database using the tablespace named 'mytablespace', type:
		`dsyjdbcmigration.bat jdbc:db2:VNURSE mytablespace`
		To migrate the subscriptions on the table DSYSAMPLE.VNMEDICALRECORD in the VNURSE database using the tablespace named 'mytablespace', type:
		`dsyjdbcmigration.bat jdbc:db2:VNURSE mytablespace dsysample.vnmedicalrecord`

`dsymigratemessagestore.bat`	This command is used by installation only when migrating. Users that have not completed their last synchronization will be reset, thus forcing a refresh on the next synchronization. Any data uploaded by the incomplete synchronization may be lost.	`dsymigratemessagestore.bat`
`dsymsgclean.bat`	For multiple server environments, this command removes all messages from the message store database.	`dsymsgclean.bat`
`dsymsgdb.bat`	This command is used by installation for multiple server environments. It creates the necessary message store database.	`dsymsgdb.bat`
`dsyreplicate.bat`	This command tells DB2 Everyplace to replicate the data located in the mirror database to the source at the first opportunity.	`dsyreplicate.bat [mirror_database]`

`dsyreset.bat`	Reset a user, device or group of users.
	For help, type `dsyreset.bat` with no arguments.
		`dsyreset.bat {[user name]} {-device [device id]} {-group [group name]} {-all}`
		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, to reset the user named 'bob', type:
		`dsyreset.bat bob`
		To reset all users belonging to the group 'Sales', type
		`dsyreset.bat -group Sales`

Table 7. Commands, for UNIX, Linux, and Windows (continued)

Command	Purpose	Usage
dsyResetUserMigration.bat	Generates an XML script to reset the users whose last synchronizations were determined to be incomplete. The input to this command takes the URL, user ID, and password for access to the control database and an absolute file name which will be the output XML file. This XML script can be run at any time. This command is normally executed during installation.	dsyResetUserMigration jdbc:db2:dsyctldb *<userID>* *<password>* *<filename.xml>* where • userID is the user ID of a DB2 Administrator • password is the password for the userID you specified • filename.xml is the output XML file
dsysetenv.bat	This script sets the environment variables specific to the DB2 Everyplace Sync Server.	dsysetenv.bat
dsysetjavahome.bat	This script command may be modified to add JDBC drivers to the classpath for MDAC. To add native libraries that these drivers may need, add those entries to dsysetenv.bat. Note that the JDBC drivers also have to be added to the application server classpath that runs the DB2 Everyplace Servlets.	
dsysetproperty.bat	Query or set a property in the control database DSY.PROPERTIES table.	Type dsysetproperty.bat without arguments for instructions.
dsysync.bat	This command launches the embedded version of the WebSphere Application Server version 5 that executes the DB2 Everyplace servlets. It will launch the server process in the background, and then exit. This command takes no arguments. If DB2 Everyplace is installed in another application server, such as WebSphere Application Server version 5, use the appropriate commands for that server.	dsysync.bat
dsysyncmonitor.bat	This command launches a GUI tool that helps view log information about users that are currently synchronizing. The GUI can be configured to update at different intervals to display the latest information.	dsysyncmonitor.bat
dsysyncstatus.bat	This command prints the execution status of the DB2 Everyplace server that is installed in the embedded version of the WebSphere Application Server version 5. The server can be started, running, or stopped. This command takes no arguments. If DB2 Everyplace is installed in another application server, such as WebSphere Application Server version 5, please use the appropriate commands for that server.	dsysyncstatus.bat

Table 7. Commands, for UNIX, Linux, and Windows (continued)

Command	Purpose	Usage

`dsysyncstop.bat`	This command stops the embedded version of the WebSphere Application Server version 5 that executes the DB2 Everyplace servlets. It will issue the stop request to the server process that is running, print the status, and then exit. This command takes no arguments.
	If DB2 Everyplace is installed in another application server, such as WebSphere Application Server version 5, use the appropriate commands for that server.
		`dsytrace [0 | 1] [-console]`
		where
		• `0` turns tracing off except for errors
		• `1` turns all tracing on
		• `-console` disables tracing to console window
		For example, to turn all tracing on and not display in console window, type:
		`dsytrace 1 -console`

`dsytraceprocess.bat`	Used to expand the timestamp format in the tracefiles to "yyyymmdd hh:mm:ss.ms" format.	`dsytraceprocess.bat -F <tracefileName> -O <true|false>` The `-O` option is used to indicate the overwrite flag. When specified to be true, the original trace file will be overwritten by the converted trace file with expanded timestamps. Otherwise the resulting trace files will be named as `originalTraceFile.new`.

`dsyunblocksubscription.bat`	This command unblocks the subscription name provided in the command argument. More than one subscription can be named at once on the command line. Unblocking a subscription that was previously blocked will instruct the Sync Server to resume accepting new changes from the clients as a part of this subscription or these subscriptions.	`dsyunblocksubscription [subscription]+`
		where
		• *`[subscription]`* is the subscription to unblock on the server.
		For example, to unblock the subscription with the name 'bob', type:
		`dsyunblocksubscription bob`
		To unblock both the subscriptions named 'bob' and 'sally', type:
		`dsyunblocksubscription bob sally`

`dsyversion.bat`	This command determines the version of the control database by querying the dsy.properties table.	`dsyversion.bat`
`useUdb8.bat`	This script sets the class path for the Mobile Devices Administration Center to be used with DB2 UDB Version 8.1.	`useUdb8.bat`
`useUdbFp7_8.bat`	This script sets the class path for the Mobile Devices Administration Center to be used with DB2 UDB Version 7.2, Fix pack 7 or later.	`useUdbFp7_8.bat`

The following table lists the scripts specific to the DB2 Everyplace Sync Server on UNIX and Linux servers.

Table 8. DB2 Everyplace Sync Server instance administration scripts

Command	Purpose	Usage

`dsyidrop.sh`	This script drops a DB2 Everyplace Sync Server instance.	`dsyidrop.sh <dsyinstance>`
		`where <dsyinstance> is the name of the DB2 Everyplace Sync Server instance to be dropped.`

Table 8. DB2 Everyplace Sync Server instance administration scripts (continued)

Command	Purpose	Usage
`dsyicreate.sh`	This script is used to create a DB2 Everyplace Sync Server instance. The DB2 Everyplace Sync Server instance must be an existing user on the system and must be a DB2 UDB instance.	To display the parameters required to create a DB2 Everyplace Sync Server instance, type : `dsyicreate.sh`
`dsyilist.sh`	This script lists the DB2 Everyplace Sync Server instances currently defined.	`dsyilist.sh`

Syntax for filters

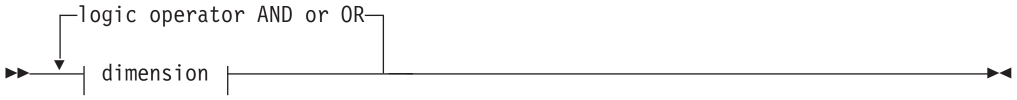
You can use a clause similar to the WHERE clause in SQL to filter for individual users when you create a replication subscription in the Mobile Devices Administration Center. Enter the clause in the **Subset of rows for individual users** field of the Advanced Replication Subscription notebook for DataPropagator subscriptions, or Advanced Replication Subscription notebook for JDBC subscriptions.

There are limitations when you use the syntax diagram listed in this topic. These limitations include:

- There can be several look-up tables (such as the Ziptab table in the example above). However, each look-up table cannot be mentioned more than once in the clause.
- Parameters can only appear where a constant is expected. If the parameter is a character data type, such as VARCHAR or CHAR, it must be enclosed in single quotes. For example, `JOBCODE=':JOB'`
- Restriction for creating filters on Lotus Domino table: In order to define a filter (Horizontal and Vertical) for a JDBC subscription using a Lotus Domino data source, the source table in the Domino database needs to be created with uppercase column names. This restriction applies to primary key columns and the columns used in the filter (columns used in the WHERE clause for a horizontal filter and columns unsubscribed for a vertical filter)

Syntax

Filter



dimension:

```
├──┬─column-name──IN──┤ subselect ├────────────────┬──────────────────────────┤
   │    ┌─,────────────┐                          │
   │    ▼              │                          │
   ├─(────column-name──┴──)──IN──┤ subselect ├────┤
   └─┤ simpleUnaryFilter ├────────────────────────┘
```

subselect:

```
├──(──┤ selectClause ├──┤ fromClause ├──┬──────────────────────┬──)──────────────┤
                                        └──┤ whereClause ├─────┘
```

selectClause:

```
               ┌─,───────────┐
               ▼             │
├──SELECT─────column-name───┴────────────────────────────────────────────────┤
```

fromClause:

```
├──FROM──┤ singleTableReference ├────────────────────────────────────────────┤
```

whereClause:

```
├──WHERE──┤ simpleFilter ├────────────────────────────────────────────────────┤
```

singleTableReference:

```
├──┬─────────────────┬──table-name───────────────────────────────────────────┤
   └─schema-name──.─┘
```

simpleFilter:

```
   ┌─logic operator AND or OR─┐
   ▼                          │
├─────┤ simpleUnaryFilter ├───┴──────────────────────────────────────────────┤
```

simpleUnaryFilter:

```
├──┬─NOT──┤ simpleUnaryFilter ├─┬────────────────────────────────────────────┤
   ├─(──┤ simpleFilter ├──)─────┤
   └─┤ simplePredicate ├────────┘
```

simplePredicate:

```
├──┤ expression ├──┤ simplePredicateEnd ├──────────────────────────────────────┤
```

expression:

```
├──┬─┤ function ├─┬───────────────────────────────────────────────────────────┤
   ├─column-name──┤
   └─┤ constant ├─┘
```

function:

```
├──┬───────────────────┬─function-name──(──┬──────────────────────┬─)──────────┤
   └─schema-name──.────┘                   │ ┌─,──────────────────┐ │
                                           │ ↓                    │ │
                                           └───┤ expression ├─────┴─┘
```

simplePredicateEnd:

```
├──┬─┬─┤ operator ├─┬─┤ expression ├──────────────────┬──────────────────┤
   │ └─IN───────────┘                                 │
   ├─IS──┬─────┬──NULL────────────────────────────────┤
   │     └─NOT─┘                                      │
   └─┬─────┬─BETWEEN──┤ expression ├──AND──┤ expression ├─┘
     └─NOT─┘
```

operator:

```
├──┬─=────┬─────────────────────────────────────────────────────────────┤
   ├─<>───┤
   ├─>────┤
   ├─<────┤
   ├─>=───┤
   ├─<=───┤
   ├─!=───┤
   ├─!<───┤
   ├─!>───┤
   └─LIKE─┘
```

Description

Operators

You can use any of the following operators:

= Equal to.

<> Not equal to.

< Less than.

> Greater than.

<= Less than or equal to.

!> Less than or equal to.

>= Greater than or equal to.

!< Greater than or equal to.

LIKE Matches one character string. Use a single-byte character-set (SBCS) underscore to refer to one SBCS character. Use a double-byte character-set (DBCS) underscore to refer to one DBCS character. For example, a condition WHERE PART_NUMBER LIKE '_0' (here the underscore is in SBCS) returns all 2–digit part numbers ending in 0 (20, 30, and 40, for example). Use a percent (either SBCS or DBCS) to refer to a string of zero or more SBCS or DBCS characters. For example, a condition WHERE DEPT_NUMBER LIKE '2%' returns all department numbers beginning with the number 2 (20, 27, or 234, for example).

Related tasks:

- "Filtering data at the subscription level" on page 43
- "Filtering data from data sources" on page 41
- "Filtering data horizontally" on page 42
- "Defining simple horizontal filters" on page 43
- "Filtering data at the group level" on page 44
- "Filtering data at the user level" on page 45
- "Defining complex horizontal filters" on page 46
- "Filtering data vertically" on page 47

XML Scripting tool DTD

This topic contains the XML Scripting DTD. For more information, see "Utilizing the XML Scripting tool" on page 131.

```
<!ELEMENT Root ( AddJdbcMaster | AddReplMaster | AddJdbcMirror |
                 AddReplMirror | AddGroup | AddUser | AddSubscriptionSet |
                 AddJdbcSubscription | AddFileSubscription |
                 AddDproprSubscription |AddUploadSubscription |
                 AddCustomSubscription | AddAdapter |
                 AddServer | ResetUser | ResetGroup | ResetAll |
                 RemoveJdbcMaster | RemoveReplMaster | RemoveJdbcMirror |
                 RemoveReplMirror | RemoveGroup | RemoveUser |
                 RemoveSubscriptionSet |RemoveSubscription |
                 RemoveAdapter | RemoveServer | AlterUser |
                 AlterGroup | AlterFileSubscription | AlterJdbcSubscription |
                 AlterUploadSubscription | AlterDproprSubscription |
                 AlterSubscriptionSet | AlterJdbcMaster | AlterJdbcMirror |
                 AlterReplMaster | AlterReplMirror | AlterAdapter )*>

<!--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:
DB2 Everyplace 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 may 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 (AddDproprSubscription 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,user can run the XML Tool to introduce this replication
subscription to sync server and create the syncrhonization infrastructure in
the mirror and the 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 Subscriptionand the Sync Server DPropR
subscription.
-->


<!-- AddJdbcMaster tag inserts the 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.-->

<!ELEMENT AddJdbcMaster (Database, Driver, UserId, Password)>


<!-- AddReplMaster 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.-->

<!ELEMENT AddReplMaster (System, Instance, Database, UserId, Password,
                                                         DatabaseAlias?)>
  <!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.-->
  <!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.
-->
```

```
<!ELEMENT AddJdbcMirror (Database, Driver, UserId, Password,
                            SyncWindow? )>
  <!ELEMENT Driver      (#PCDATA)>
  <!ELEMENT SyncWindow  (#PCDATA|NULL)*>


<!-- AddReplMirror 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.

<!--Users and groups in DB2Everyplace 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. In case of LDAP enabled installations, XML tool ignores
add user/group and remove user/group tags and such user/group maintenance is not
done by the Sync Server.
-->


<!--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 'Constrnt' 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 may override this).

The data to be synchronized may 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
containSubscriptions 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 may 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. If the need arises, 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
  MIDP
  LINUX         or LINUX_X86
  EMBEDDEDLINUX or EMBEDDEDLINUX_X86
  EPOC          or EPOC_ARM
  EPOCUNICODE   or EPOCUNICODE_ARM

We have also defined a set of other possible variants based on device
processor, howeverthese 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

'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 fromthe 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*, E-mail?,
PreferredLanguage?, Interest?)>
 <!ELEMENT FirstName (#PCDATA)>
 <!ELEMENT LastName (#PCDATA)>
 <!ELEMENT AddToSyncGroup (#PCDATA)>
 <!ELEMENT RemoveFromSyncGroup (#PCDATA)>
 <!ELEMENT E-mail (#PCDATA)>
 <!ELEMENT PreferredLanguage (#PCDATA)>
 <!ELEMENT Interest (#PCDATA)>


<!--A subscription set is a logical grouping of subscriptions that are
synchronized together.A group of users synchronize 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
includedin 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
relationaldata from a mirror database to the mobile databases.
Replication from master to mirror will also be handled by the
Sync Server replication mechanism 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/orreplicated.

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". If it is not specified,
it defaults to "TRUE". After creating a subscription, a replication is
performed to fill the new mirrortables 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 mutltiple subscriptions, for
performance, it may be desireable 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 "dsyreplicate" 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 setup of DPropR replication
subscription must be done using DPropR toolsbefore the Sync Server subscription
can be created. While creating the replication subscription using DB2 DPropR tools,
use the same subscription name as the Sync Server subscription you intend
to use. Also, for Apply Qualifier, use the mirror database name.
The difference between JDBC subscriptions and the DPropR subscriptions is the
type of replication employed. Since 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
databasedefinition that is typically created by AddReplMirror tag.
If such description does not exist, creation of this subscription fails.
Sync Server assumes thatApply Qualifier of a replication subscription is
the same as this 'MirrorDb' name.
```

```
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, (AddTable*))>

  <!ATTLIST AddDproprSubscription Replicate (TRUE | FALSE) "TRUE">

  <!ELEMENT MasterSystem    (#PCDATA)>
  <!ELEMENT MasterInstance  (#PCDATA)>
  <!ELEMENT ApplyQualifier  (#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)*>

  <!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 incase 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 priviledges does not cause the table to be dropped
and recreated on themirror 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 may include data filter parameters defined for groups (see AddDataFilter tag). If data filters are defined, they will be replaced by the values specificto the user or the group of the user being synchronized. Changing this whereclause causes the table to be dropped and recreated on the target devices and the mirrordatabase.

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. Since 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 its 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.

'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 than 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 encrytion 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.


<!ELEMENT AddTable (TargetSchema, TargetTable, SourceSchema, SourceTable,
                      DeleteAuth, InsertAuth, UpdateAuth, SelectAuth,
                      WhereClauseMirrorToMobile?, WhereClauseMasterToMirror?,
                      JoinFilter?, TableSpace?, Order?, Encrypt?,
                      (AddColumn | AddIndex)*)>
  <!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)>


<!--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.
ForJDBC 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.

'Subscribe' is either 'Y' or 'N'. Default is 'N'. When 'Y',
it means column willbe synchronized.

'Replicate' is either 'Y' or 'N'. Default is 'N'. When 'Y',
it means column willbe replicated between master and mirror databases.
You cannot synchronize a column without replicating it.
```

```
'Key' is either 'Y' or 'N'. Default is 'N'. It indicates if
this column is part of a unique key constraint.

'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 contstraint 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. Since such a database
belongs to a particular 'device user', this data filter can contain
Data Filters created for groups and overriden by users
(Please see AddDataFilter and OverrideDataFilter tags).
For example, to create a column such that it's 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 polymophically 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.
-->
<!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)>


<!--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 may 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)>


<!--'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 may 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 may 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)>


<!--
  'RemoveJdbcMirror' removes records of a JDBC mirror database system
   that may 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 may 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)>


<!--'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 implemented as ResetDevice on its own, so it is not
   found in here.-->


<!--'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 may need manual cleanup after
   that. This is useful in test/development environments where developer
   does not care about source and mirror databases or they may already
   be dropped.-->
  <!ATTLIST RemoveSubscription StopOnError (TRUE | FALSE) "TRUE">


<!--'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>ausername</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 may 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.
-->
<!ELEMENT AlterJdbcMaster (Database, Driver, JdbcMaster)>
<!ELEMENT JdbcMaster (Database?, Driver?, UserId?, Password?)>

<!--'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?*)>

<!--'AlterReplMaster' allows editing of existing DPropR master database
information.-->
<!ELEMENT AlterReplMaster (System, Instance, Database, ReplMaster)>
<!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)>
<!ELEMENT ReplMirror (Database?, UserId?, Password?, SyncWindow?*)>


<!--'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 resent to the devices.
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?,
                             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)* )>

<!-- LDAPProperties tag is sufficient
<!ELEMENT AlterLDAPProperties (FirstName?, LastName?, E-mail?,
                                              PreferredLanguage?, Interest?)>
-->

<!--'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 syncrhonization 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)>
<!ELEMENT DproprSubscription (Name?, Description?,
                              EncryptionLevel?, Block?,
                              (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 | RemoveIndex)*)>

<!ELEMENT RemoveIndex (Name)>
```

Related concepts:

- "Utilizing the XML Scripting tool" on page 131

Data type mapping and restrictions

This chapter provides information about restrictions on data mapping between DB2 Everyplace and other database products. The topics covered are:

- "Data type mappings between DB2 Everyplace and data sources"
- "DB2 Universal Database" on page 221
- "Informix" on page 222
- "Oracle" on page 223
- "Sybase" on page 224
- "Lotus Domino" on page 225
- "SQL Server 2000" on page 226
- "Cloudscape" on page 226
- "Data type mapping restrictions" on page 227

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. The data source mappings listed in this family of topics include:

- DB2 Universal Database
- Informix
- Oracle
- SQL Server Type
- Sybase
- Lotus Domino
- Data type mapping restrictions
- Cloudscape

Notes:

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 may not be possible.
2. If a data type is not listed in the data type mapping tables, it is not supported.

DB2 Universal Database

Table 9 lists the data type mapping that is performed when the source data type is a DB2 Universal Database data type.

Table 9. Data type mapping from DB2 Universal Database source data types

DB2 UDB source data type	DB2 UDB mirror data type	DB2 Everyplace device data type	Cloudscape 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 VARTING
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

Related reference:

- “Data type mappings between DB2 Everyplace and data sources” on page 220
- “Informix”
- “Oracle” on page 223
- “Sybase” on page 224
- “Lotus Domino” on page 225
- “Cloudscape” on page 226

Informix

Table 10 lists the data type mapping that is performed when the source data type is an Informix data type.

Table 10. Data type mapping from Informix source data types

Informix source data type	DB2 UDB mirror data type	DB2 Everyplace device data type	Cloudscape 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 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

Table 10. Data type mapping from Informix source data types (continued)

Informix source data type	DB2 UDB mirror data type	DB2 Everyplace device data type	Cloudscape device data type
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 220
- "DB2 Universal Database" on page 221
- "Oracle"
- "Sybase" on page 224
- "Lotus Domino" on page 225
- "Cloudscape" on page 226

Oracle

Table 11 lists the data type mapping that is performed when the source data type is an Oracle data type.

Table 11. Data type mapping from Oracle source data types

Oracle 8.1.7source data type	DB2 mirror data type	DB2 Everyplace device data type	Cloudscape 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

Table 11. Data type mapping from Oracle source data types (continued)

Oracle 8.1.7source data type	DB2 mirror data type	DB2 Everyplace device data type	Cloudscape device data type
UROWID	unsupported	unsupported	unsupported
VARCHAR2(n)	VARCHAR	VARCHAR	VARCHAR
TIMESTAMP	TIMESTAMP	TIMESTAMP	TIMESTAMP

- “Data type mappings between DB2 Everyplace and data sources” on page 220
- “DB2 Universal Database” on page 221
- “Informix” on page 222
- “Sybase”
- “Lotus Domino” on page 225
- “Cloudscape” on page 226

Sybase

Table 12 lists the data type mapping that is performed when the source data type is a Sybase data type.

Table 12. Data type mapping from Sybase source data types

Sybase source data type	DB2 mirror data type	DB2 Everyplace device data type	Cloudscape device data type
BINARY(n)	unsupported	unsupported	unsupported
BIT	SMALLINT	SMALLINT	SMALLINT
CHAR(n)	CHARACTER	CHARACTER	CHARACTER
DATETIME	TIMESTAMP	TIMESTAMP	TIMESTAMP
DECIMAL[(p[,s])]	DECIMAL	DECIMAL	DECIMAL
DOUBLE PRECISION	FLOAT	VARCHAR	DOUBLE PRECISION
FLOAT(p)	FLOAT	VARCHAR	DOUBLE PRECISION
IMAGE	LONG VARCHAR FOR BIT DATA	BLOB	LONG BIT VARYING
INT	INTEGER	INT	INT
MONEY	DECIMAL	DECIMAL	DECIMAL
NCHAR(n)	CHARACTER	CHARACTER	CHARACTER
NUMERIC(p,s)	NUMERIC	DECIMAL	NUMERIC
NVARCHAR(n)	VARCHAR	VARCHAR	VARCHAR
REAL	REAL	VARCHAR	REAL
SMALLDATETIME	TIMESTAMP	TIMESTAMP	TIMESTAMP
SMALLINT	SMALLINT	SMALLINT	SMALLINT
SMALLMONEY	DECIMAL	DECIMAL	DECIMAL
TEXT	LONG VARCHAR	VARCHAR	LONG VARCHAR
TINYINT	unsupported	unsupported	unsupported
VARBINARY(n)	VARCHAR() FOR BIT DATA	BLOB	BIT VARYING
VARCHAR(n)	VARCHAR	VARCHAR	VARCHAR

Related reference:

- "Data type mappings between DB2 Everyplace and data sources" on page 220
- "DB2 Universal Database" on page 221
- "Informix" on page 222
- "Oracle" on page 223
- "Lotus Domino"
- "Cloudscape" on page 226

Lotus Domino

Table 13 lists the data type mapping that is performed when the source data type is a Lotus Domino type.

Table 13. Data type mapping from Lotus Domino source data types

Domino	DB2 UDB mirror data type	DB2 Everyplace device data type
CHAR(n)	CHARACTER	CHARACTER
DATE	DATE	DATE
DECIMAL[(p[,s])]	DECIMAL	DECIMAL
DOUBLE PRECISION	DOUBLE	VARCHAR
FLOAT(p)	DOUBLE	VARCHAR
INTEGER	DOUBLE	VARCHAR
LONGVARCHAR	LONGVARCHAR	VARCHAR
NUMERIC(p,s)	DECIMAL	DECIMAL
REAL	DOUBLE	VARCHAR
SMALLINT	SMALLINT	SMALLINT
TIME	TIME	TIME
TIMESTAMP	TIMESTAMP	TIMESTAMP
VARCHAR(n)	VARCHAR	VARCHAR
BIGINT	N/A	N/A
BINARY	N/A	N/A
BIT	N/A	N/A
LONGVARBINARY	N/A	N/A
TINYINT	N/A	N/A
VARBINARY	N/A	N/A

Related reference:

- "Data type mappings between DB2 Everyplace and data sources" on page 220
- "DB2 Universal Database" on page 221
- "Informix" on page 222
- "Oracle" on page 223
- "Sybase" on page 224

SQL Server 2000

Table 14 lists the data type mapping that is performed when the source data type is SQL Server 2000.

Table 14. Data type mapping from SQL Server 2000

SQL Server 2000 source type	DB2 UDB mirror data type	DB2 Everyplace device data type	Cloudscape 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 220
- “DB2 Universal Database” on page 221
- “Informix” on page 222
- “Oracle” on page 223
- “Lotus Domino” on page 225

Cloudscape

Table 15 on page 227 lists the data type mapping that is performed when the source data type is a Cloudscape data type.

Table 15. Data type mapping from Cloudscape source data types

Cloudscape source data type	DB2 mirror data type	DB2 Everyplace device data type	Cloudscape device data type
BIGINT	BIGINT	VARCHAR,	BIGINT
BIT	unsupported	unsupported	unsupported
BIT VARYING	LONG VARCHAR FOR BIT DATA	BLOB	BIT VARYING
BOOLEAN	BOOLEAN	SMALLINT	SMALLINT
CHAR	CHAR	CHAR	CHAR
DATE	DATE	DATE	DATE
DECIMAL	DECIMAL	DECIMAL	DECIMAL
DOUBLE PRECISION	FLOAT	VARCHAR,	DOUBLE PRECISION
FLOAT	FLOAT	VARCHAR,	FLOAT
INT	INT	INT	INT
LONG NVARCHAR	LONG VARCHAR	VARCHAR	LONG VARCHAR
LONG BIT VARYING	LONG BIT VARYING	BLOB	LONG BIT VARYING
NATIONAL CHAR	unsupported	unsupported	unsupported
NATIONAL CHAR VARYING	unsupported	unsupported	unsupported
NUMERIC	NUMERIC	DECIMAL	NUMERIC
REAL	REAL	VARCHAR	REAL
SMALLINT	SMALLINT	SMALLINT	SMALLINT
TIME	TIME	TIME	TIME
TIMESTAMP	TIMESTAMP	TIMESTAMP	TIMESTAMP
TINYINT	SMALLINT	SMALLINT	SMALLINT
VARCHAR	VARCHAR	VARCHAR	VARCHAR

Related reference:

- “Data type mappings between DB2 Everyplace and data sources” on page 220
- “DB2 Universal Database” on page 221
- “Informix” on page 222
- “Oracle” on page 223
- “Sybase” on page 224
- “Lotus Domino” on page 225

Data type mapping restrictions

The following restrictions exist when data type mapping is performed:

- If DB2 UDB mirror types GRAPHIC, VARGRAPHIC are used, the DB2 UDB 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.

- Sybase NCHAR can be used in single-byte languages only
- 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 Sync Server:
  - **DB2 UDB Type:** LONG VARCHAR, LONG VARCHAR FOR BIT DATA, LONG VARGRAPHIC, and VARCHAR() FOR BIT DATA
  - **Informix 9.x Type:** DATETIME, and INTERVAL
  - **Oracle 8.1.7 Type:** RAW
  - **SQL Server 2000 Type:** MONEY, REAL, and SMALLMONEY
  - **Sybase 11.9.2 Type:** BIT, DOUBLE PRECISION, FLOAT, IMAGE, MONEY, SMALLMONEY, TEXT, and VARBINARY
  - **Lotus Domino Type:** LONG VARCHAR

Part 5. Appendixes

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
U.S.A.

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 Canada Limited
Office of the Lab Director
1150 Eglinton Ave. East
North York, Ontario
M3C 1H7
CANADA

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.

ACF/VTAM
AISPO
AIX
AIX/6000
AIXwindows
AnyNet
APPN
AS/400
BookManager
CICS
C Set++
C/370
DATABASE 2
DataHub
DataJoiner
DataPropagator
DataRefresher
DB2
DB2 Connect
DB2 Extenders
DB2 OLAP Server
DB2 Universal Database
Distributed Relational Database Architecture
DRDA
eNetwork
Extended Services
FFST
First Failure Support Technology
IBM
IMS
IMS/ESA
iSeries
LAN DistanceMVS
MVS/ESA
MVS/XA
Net.Data
OS/2
OS/390
OS/400
PowerPC
QBIC
QMF
RACF
RISC System/6000
RS/6000
S/370
SP
SQL/DS
SQL/400
System/370
System/390
SystemView
VisualAge
VM/ESA
VSE/ESA
VTAM
WebExplorer
WIN-OS/2
z/OS

The following terms are trademarks or registered trademarks of other companies:

Microsoft, Windows, and Windows NT are trademarks or registered trademarks of Microsoft Corporation.

Java or all Java-based trademarks and logos, and Solaris are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Tivoli and NetView are trademarks of Tivoli Systems Inc. in the United States, other countries, or both.

UNIX is a registered trademark in the United States, other countries or both and is licensed exclusively through X/Open Company Limited.

Other company, product, or service names, which may be denoted by a double asterisk(**) may be trademarks or service marks of others.

Glossary

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 administration control database to ensure that the user is authorized to use the 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 Administrator.

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 Administrator object containing synchronization error messages and their descriptions.

M

master database. See *source database*.

MDAC. See *Administrator*.

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 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.

Administrator (MDAC). A graphical interface that allows you to create, edit, and view synchronization objects and their relationships to each other. The Administrator also allows you to view synchronization status of individual clients and error messages.

O

object.

1. Anything that can be created or manipulated with SQL—for example, tables, views, indexes, or packages.
2. 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

QBE. See *Query-by-Example*.

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.

Query-by-Example. An application that allows a user to dynamically view and modify the data stored in a DB2 Everyplace table.

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 Administrator 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 Administrator 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.

Index

Contacting IBM

For information or to order any of the DB2 Everyplace products contact an IBM representative at a local branch office or contact any authorized IBM software remarketer.

If you live in the U.S.A., then you can call one of the following numbers:

- 1-800-237-5511 for customer support
- 1-888-426-4343 to learn about available service options

Product Information

If you live in the U.S.A., then you can call one of the following numbers:

- 1-800-IBM-CALL (1-800-426-2255) or 1-800-3IBM-OS2 (1-800-342-6672) to order products or get general information.
- 1-800-879-2755 to order publications.

http://www.ibm.com/software/data/db2/everyplace/
The DB2 Everyplace World Wide Web pages provide current DB2 Everyplace information about news, product descriptions, education schedules, and more.

http://www.ibm.com/software/data/db2/everyplace/library.html
The DB2 Everyplace Technical Library provides access to frequently asked questions, fixes, books, and up-to-date DB2 Everyplace technical information.

Note: This information may be in English only.

http://www.ibm.com/software/data/
The DB2 World Wide Web pages provide current DB2 information about news, product descriptions, education schedules, and more.

http://www.ibm.com/software/data/db2/library/
The DB2 Product and Service Technical Library provides access to frequently asked questions, fixes, books, and up-to-date DB2 technical information.

Note: This information may be in English only.

http://www.elink.ibmlink.ibm.com/pbl/pbl/
The International Publications ordering Web site provides information on how to order books.

http://www.ibm.com/education/certify/
The Professional Certification Program from the IBM Web site provides certification test information for a variety of IBM products, including DB2.

ftp.software.ibm.com
Log on as anonymous. In the directory `/ps/products/db2`, you can find demos, fixes, information, and tools relating to DB2 and many other products.

comp.databases.ibm-db2, bit.listserv.db2-l
These Internet newsgroups are available for users to discuss their experiences with DB2 products.

On Compuserve: GO IBMDB2

Enter this command to access the IBM DB2 Family forums. All DB2 products are supported through these forums.

For information on how to contact IBM outside of the United States, refer to Appendix A of the *IBM Software Support Handbook*. To access this document, go to the following Web page: http://www.ibm.com/support/, and then select the IBM Software Support Handbook link near the bottom of the page.

Note: In some countries, IBM-authorized dealers should contact their dealer support structure instead of the IBM Support Center.

Program Number: 5724–D04

Printed in USA

SC18-7186-01