

# Mobilizing a Sybase Adaptive Server Enterprise 15 Database

A whitepaper from Sybase iAnywhere

## TABLE OF CONTENTS

1	Data mobilization: An overview
1	Prerequisites
2	Synchronization environment
2	Consolidated schema
4	Preparing the consolidated database
5	Connecting with MobiLink
6	Creating a synchronization model
10	Deploying the synchronization model
14	Synchronizing
14	Starting the consolidated database
14	Starting the MobiLink server
14	Starting the remote database
15	Initializing the remote ID
16	Synchronizing the remote client
16	Viewing remote client data in Sybase Central
17	Conclusion
18	Legal Notice
18	Contact Us

## DATA MOBILIZATION: AN OVERVIEW

Data mobilization provides a way for enterprises to bring their mission-critical information to personnel at the front lines of business. It enables employees at remote or branch offices to access and manipulate corporate data without requiring a constant network connection to headquarters. This significantly reduces connectivity costs and issues, and ensures that remote office workers are productive because they always have access to data.

MobiLink is a session-based synchronization system that allows two-way synchronization between a main database, called the consolidated database, and many remote databases. The consolidated database, which can be one of several ODBC-compliant databases, holds the master copy of all the data. Remote databases can be either SQL Anywhere or UltraLite databases. MobiLink synchronization is a component of SQL Anywhere, the comprehensive package for mobile data management from iAnywhere Solutions.

Synchronization typically begins when a MobiLink remote client opens a connection to a MobiLink server. During synchronization, the MobiLink client uploads database changes that were made to the remote database since the previous synchronization. On receiving this data, the MobiLink server updates the consolidated database, and then downloads changes on the consolidated database to the remote database.

MobiLink is proven to be a scalable, high-performance data synchronization technology, placing iAnywhere as a significant leader in the mobile workspace. It functions on multiple platforms (including Windows, Mac OS X, Linux, Pocket PC, Windows Mobile 5.0, Symbian, and Palm) and works across any type of wired or wireless connection. In terms of development and database design, there is the option of the intuitive and easy-to-use graphical interface of Sybase Central, or alternatively for more advanced developers, there is the option to configure the SQL code directly. This allows the developer to concentrate on their code. A wide variety of development technologies are supported, including .NET and Java. Other features include monitoring and reporting, performance tuning, and security.

For more information on each of the components in a MobiLink synchronization system, you should consult the product documentation. The section entitled "Introducing MobiLink Synchronization" in the book *MobiLink - Getting Started* gives an excellent overview. It is recommended that you read this section before proceeding with this whitepaper.

SQL Anywhere documentation is available online at  
[http://www.iAnywhere.com/developer/product\\_manuals/sqlanywhere/](http://www.iAnywhere.com/developer/product_manuals/sqlanywhere/).

The rest of this document walks through the mobilization of an Adaptive Server Enterprise database.

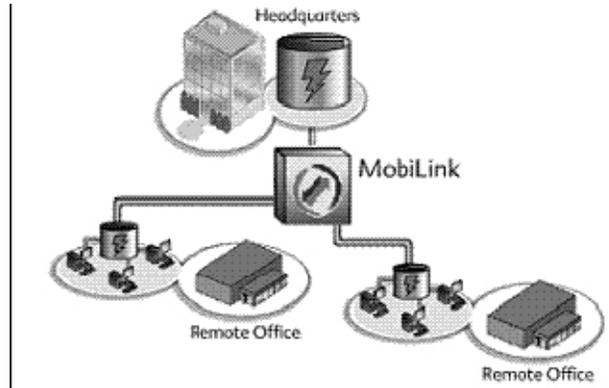
## PREREQUISITES

- Adaptive Server Enterprise 15.0
- SQL Anywhere 10.0.1
- MobiLink 10 (included in SQL Anywhere 10)

This demonstration uses the pubs2 sample schema that is provided with Adaptive Server Enterprise 15.0. This schema is an optional part of the Adaptive Server Enterprise install. Information about this sample can be found in the Adaptive Server Enterprise documentation or can be viewed online at [http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.help.ase\\_15.0-sag1/html/sag1/sag150.htm](http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.help.ase_15.0-sag1/html/sag1/sag150.htm).

## SYNCHRONIZATION ENVIRONMENT

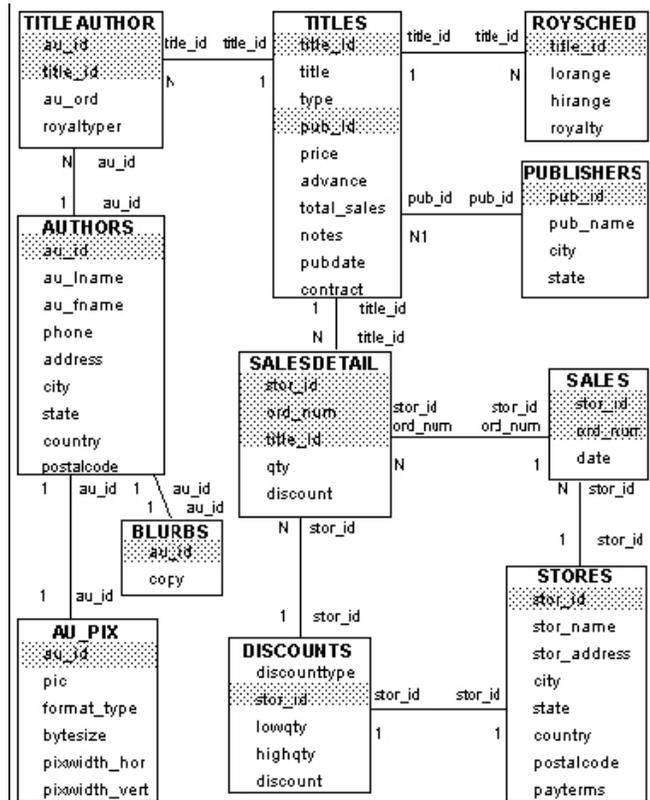
The purpose of this demonstration is to mobilize data pertaining to a chain of bookstores. The MobiLink server is capable of synchronizing with SQL Anywhere or Ultralite clients. In this scenario, each remote client is a SQL Anywhere database running at a branch office. The following diagram illustrates the synchronization environment.



Each branch office in this scenario is a remote synchronization environment. Each remote office has a local SQL Anywhere database that is synchronized with the corporate Adaptive Server Enterprise database. Each office can have several computers that access and manipulate data from the remote database.

## CONSOLIDATED SCHEMA

The consolidated database schema encapsulates information about stores, titles, authors, publishers, and sales. The following diagram shows the relationships between the different objects in the schema.



Each object in the schema diagram represents a table that exists in a consolidated Adaptive Server Enterprise database:

Table	Description
AUTHORS	The authors of the various TITLES in the system.
AU_PIX	Pictures of the authors.
DISCOUNTS	Records of various discounts at particular STORES.
SALES	Each sale record is one sale made by a particular store.
SALESDetail	Contains information about the different TITLES that were included in a particular sale
TITLES	Records of all of the different books in the system.
TITLEAUTHOR	Contains information about which TITLES were written by which AUTHORS.
STORES	Each store record is one store or branch office in the system.
ROYSCHED, PUBLISHERS, and BLURBS	Contain information that is not needed in this demonstration.

This walkthrough explains how to make a subset of this data available locally to each store in the chain of bookstores.

### Designing the remote schema

The first step is to design a remote schema. It is unnecessary and inefficient for each store to have a copy of the entire consolidated database. The remote schema is designed so that it only contains information relevant to one particular store. To achieve this, it is designed as a subset of the consolidated database in the following way:

Consolidated table	Remote table
TITLEAUTHOR	Includes all rows.
AUTHORS	Includes all rows.
BLURBS	Not included on remote.
AU_PIX	Includes all rows.
TITLES	Includes all rows.
ROYSCHED	Not included on remote.
PUBLISHERS	Not included on remote.
STORES	Filter by stor_id.
DISCOUNTS	Filter by stor_id.
SALES	Filter by stor_id.
SALESDetail	Filter by stor_id.

Each store needs to keep records of all titles and authors, so customers can come into their store and search their inventory by the title or author of the book. However a bookstore does not need information regarding publishers or royalties associated with each title, so this information is not synchronized to each store. Each store needs information about sales and discounts, but not about sales and discounts related to other stores. This is achieved by

filtering rows based on a store identifier.

**Note**

The table above demonstrates designing a remote schema by excluding tables and taking a subset of rows from other tables. You can also take a subset of columns from a table if certain columns are not required on the remotes.

The next step is to choose the synchronization direction of each table. What is important to consider here is what information a remote needs to read (select), and what information a remote needs to create (insert), change (update), or remove (delete). In this example, a specific bookstore needs a list of authors and titles, but never enters a new author into the system. This places a restriction that authors and titles must always enter the system from the consolidated database at the headquarters. However, a bookstore needs to be able to record new sales on a regular basis. These factors lead to the following decisions about each table:

Table	Synchronization
TITLEAUTHOR	Download to remote only.
AUTHORS	Download to remote only.
AU_PIX	Download to remote only.
TITLES	Download to remote only.
STORES	Download to remote only.
DISCOUNTS	Download to remote only.
SALES	Download and upload.
SALESDetail	Download and upload.

**PREPARING THE CONSOLIDATED DATABASE**

In a Mobilink synchronization system, the primary key of a table is the only way to identify the same row in different databases and the only way to detect conflicts. As such, every table that is being mobilized must have a primary key and must not be updated. Note that the sample database pubs2 was not designed with primary keys, but instead uses unique indexes on each table. You can alter each table to use primary keys instead of unique indexes by running the following data definition language (DDL) commands against the pubs2 database.

```
DROP INDEX au_pix.au_pixind
DROP INDEX authors.authosind
DROP INDEX sales.salesind
DROP INDEX salesdetail.salesdetailind
DROP INDEX titleauthor.taind
DROP INDEX titles.titleidind
DROP INDEX discounts.stor_id
```

```
ALTER TABLE au_pix ADD PRIMARY KEY (au_id)
ALTER TABLE authors ADD PRIMARY KEY (au_id)
ALTER TABLE sales ADD PRIMARY KEY (stor_id, ord_num)
ALTER TABLE salesdetail ADD detail_id NUMERIC(8,0) IDENTITY
ALTER TABLE salesdetail ADD PRIMARY KEY(detail_id, stor_id)
ALTER TABLE titleauthor MODIFY au_ord not null
ALTER TABLE titleauthor ADD PRIMARY KEY( au_id, title_id, au_ord)
ALTER TABLE titles ADD PRIMARY KEY (title_id)
ALTER TABLE stores ADD PRIMARY KEY (stor_id)
ALTER TABLE discounts ADD PRIMARY KEY (stor_id)
```

After running these DDL commands, the MobiLink server should have no trouble connecting to the database and setting it up for synchronization with any number of remotes.

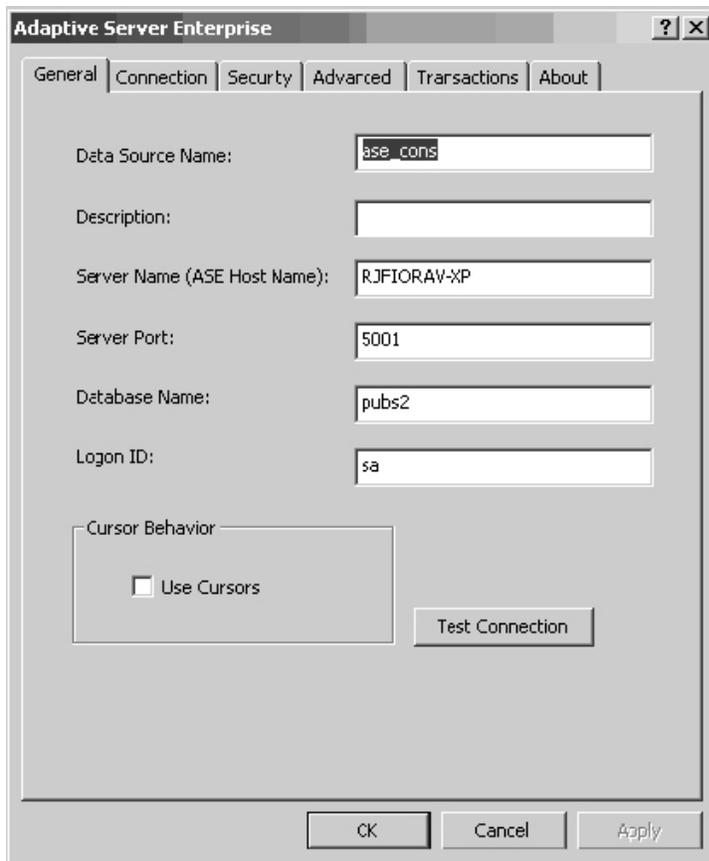
**Note**

It is possible to synchronize data with consolidated databases that do not have primary keys. However, you must write your own synchronization events that act on shadow tables that are designed to identify rows uniquely in other tables.

**CONNECTING WITH MOBILINK**

1. Verify that your Adaptive Server Enterprise 15 service is running and that the pubs2 database is configured properly.
2. Create an ODBC data source that points to pubs2.

You should use the ODBC driver provided by Adaptive Server Enterprise, and the configuration should look similar to the image below.

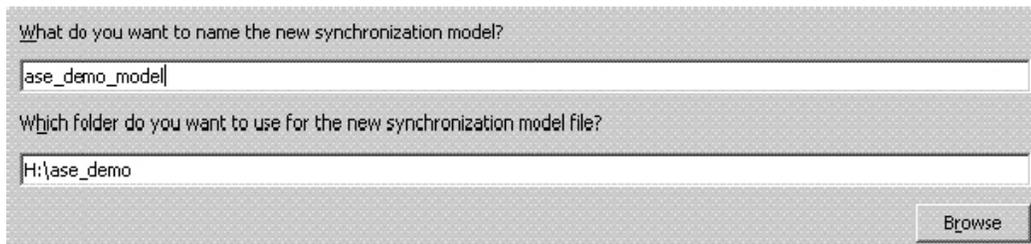


After configuring your ODBC data source, you can use the MobiLink plug-in to connect to the Adaptive Server Enterprise database and create a synchronization model.

<sup>1</sup> For more information about recommended ODBC drivers for MobiLink, see [http://www.ianywhere.com/developer/technotes/odbc\\_mobilink.html](http://www.ianywhere.com/developer/technotes/odbc_mobilink.html).

## CREATING A SYNCHRONIZATION MODEL

1. Start Sybase Central by choosing Programs ► SQL Anywhere 10 ► Sybase Central.
2. From the Tools menu, choose MobiLink 10 ► Set Up MobiLink Synchronization. The Create Synchronization Model wizard appears.
3. Enter the name and location of your new model, and then click Next.



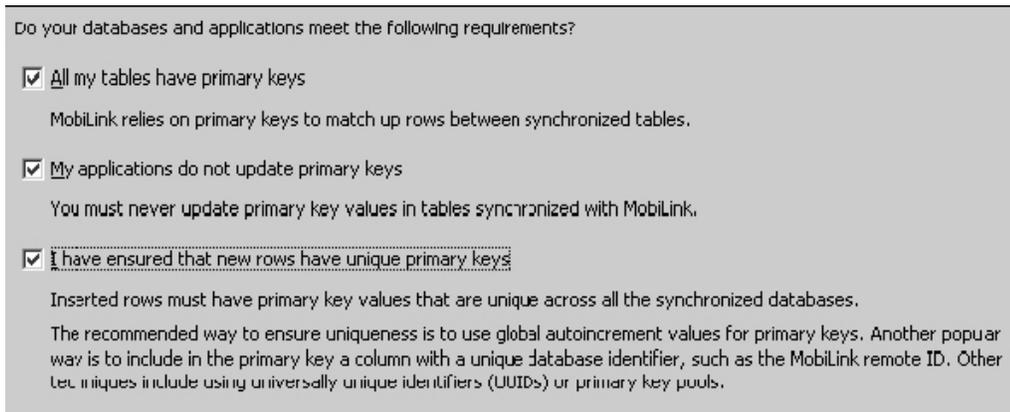
What do you want to name the new synchronization model?  
ase\_demo\_model

Which folder do you want to use for the new synchronization model file?  
H:\ase\_demo

Browse

4. The next page verifies that the remote database schema is ready for synchronization. You must select all of the options on this page before you can proceed.

Since you previously altered pubs2 to use primary keys instead of unique indexes, select all of the options, and then click Next.



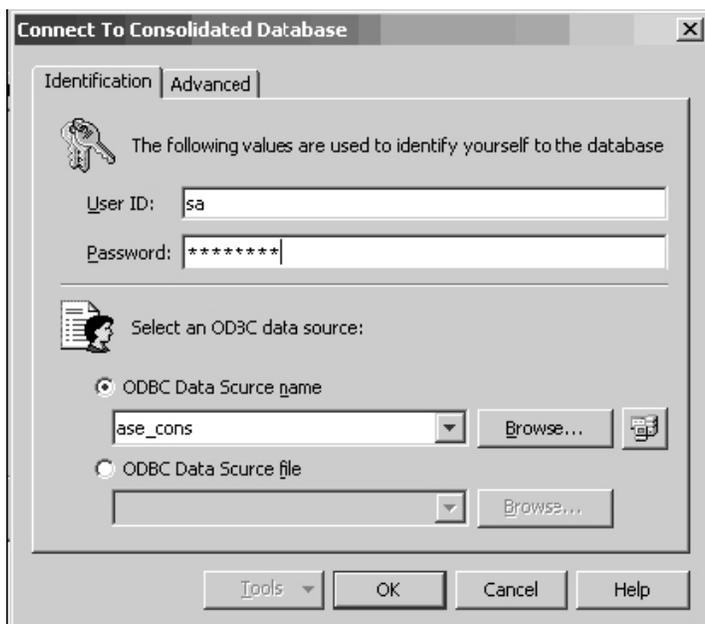
Do your databases and applications meet the following requirements?

All my tables have primary keys  
MobiLink relies on primary keys to match up rows between synchronized tables.

My applications do not update primary keys  
You must never update primary key values in tables synchronized with MobiLink.

I have ensured that new rows have unique primary keys  
Inserted rows must have primary key values that are unique across all the synchronized databases.  
The recommended way to ensure uniqueness is to use global autoincrement values for primary keys. Another popular way is to include in the primary key a column with a unique database identifier, such as the MobiLink remote ID. Other techniques include using universally unique identifiers (UUIDs) or primary key pools.

5. Choose the consolidated database.  
Type the name of the data source you created that points to pubs2, the user ID, and the password. Click OK.



Connect To Consolidated Database

Identification | Advanced

The following values are used to identify yourself to the database

User ID: sa

Password: \*\*\*\*\*

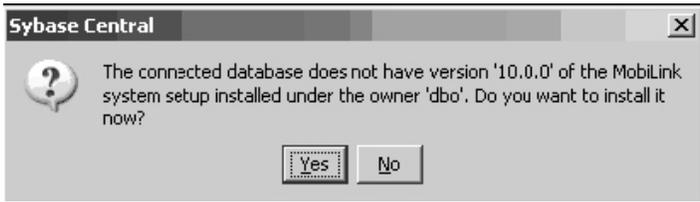
Select an ODBC data source:

ODBC Data Source name  
ase\_cons Browse... 

ODBC Data Source file  
Browse...

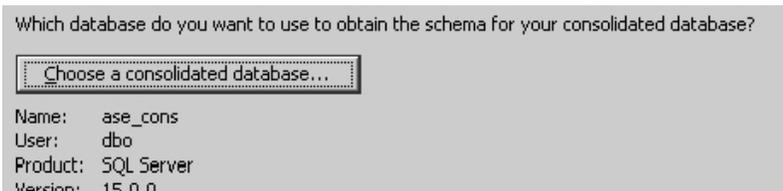
Tools OK Cancel Help

6. If this is the first time your database has been used by MobiLink, the following dialog appears.



Click Yes. MobiLink automatically creates all the tables necessary for the database to be used in a synchronization environment. This does not alter your existing schema; it only adds MobiLink system tables and procedures.

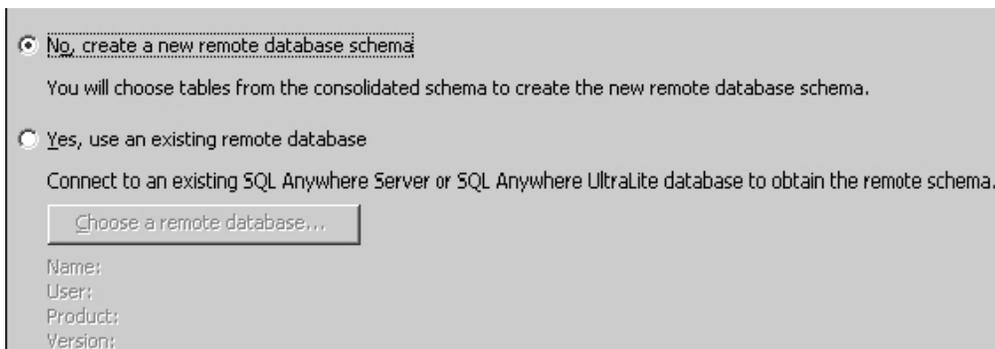
7. If the connection and system setup was successful you see a page similar to this:



If you see an error message, ensure that the data source is configured properly.

8. Choose the remote schema.

Since you have not created a remote schema yet, have MobiLink create one based on the consolidated schema by selecting No, Create a New Remote Database Schema.



9. Select the checkboxes for the following tables to include them in the remote schema, and then click Next.

- authors
- discounts
- sales
- salesdetail
- stores
- titleauthor
- titles

10. Select Timestamp-based Download.

Choosing timestamp-based downloads minimizes the amount of data that is transferred because only data that has been updated since the last download is transmitted.

**Timestamp-based download**  
Only download rows that have changed since the last download.

**Snapshot download**  
Download all rows every synchronization, even if they have been previously downloaded.

**Custom download logic**  
You will write your own `download_cursor` and `download_delete_cursor` scripts instead of having them generated automatically. You can write them in the Events editor.

11. Choose how to implement timestamp-based downloads. On this page, select Use Shadow Tables to Hold Timestamp Columns. Using shadow tables is often preferred because it doesn't require any changes to existing tables. Choosing the first option allows Mobilink to add a column to every synchronized table in the consolidated schema.

The download-by-timestamp option requires a timestamp column for each consolidated table to track when rows are modified.

Timestamp column name:

Use timestamp columns in synchronized tables  
The column will be added to each synchronized consolidated table if it does not already exist, and it will be excluded from synchronization. Use this option to avoid joining with a shadow table when selecting the data to download.

**Use shadow tables to hold timestamp columns**  
A shadow table will be created for each synchronized consolidated database table. Use a shadow table if you do not want to modify the consolidated schema.

12. Configure how to propagate record deletions to remote devices. Select Yes to indicate that you want the remotes to download deletes. Select Use Shadow Tables to Record Deletions. Mobilink creates shadow tables on the consolidated database to implement this.

Do you want data deleted on the consolidated database to be deleted on the remote databases?

**Yes**  No

Since you are not using snapshot download, you need to record which rows have been deleted.

How do you want to record row deletions?

**Use shadow tables to record deletions**  
A shadow table will be created to hold the primary key and time of deletion for each deleted row.

Timestamp column name:

Use logical deletes  
Instead of deleting rows in the consolidated database, your applications will use a one-character column to indicate if the row has been deleted. This column will be added if not already present.

13. Specify that the remotes download only a subset of data from the consolidated database.

Do you want all remote databases to download the same data?

**Yes, download the same data to each remote**

No, download different subsets to different remote databases

Although the data will be partitioned by remote database, you will specify how later by using custom logic for each table in Model mode.

14. Choose No Conflict Detection.

For most applications you want to select either row or column-based conflict detection to ensure consistency throughout remotes, but for the purposes of this tutorial, you can select no conflict detection.

Which type of conflict detection do you want?

**No conflict detection**  
Always apply uploaded updates without checking for conflicts (best performance).

**Row-based conflict detection**  
A conflict is detected if the row has been updated on both the remote and the consolidated databases since the last synchronization.

**Column-based conflict detection**  
A conflict is detected only if the same column has been updated for the row in both the remote and consolidated databases. Otherwise only the uploaded column updates will be applied.  
Note: If a table has BLOB columns then row-based conflict detection will be used.

15. Name your publication and script version, and then click Next.

The publication is the object that specifies what data is to be synchronized. The script defines how data that is uploaded from remotes should be applied to the consolidated database. You can use different script versions for different applications, allowing you to maintain a single MobiLink server while synchronizing to entirely different applications.

The remote tables that you have chosen to synchronize will be grouped into a publication, and the synchronization logic scripts will be grouped into a script version.

What do you want to name the publication?

sync\_ase\_publication

What do you want to name the script version?

sync\_ase\_script

16. You should now be in the Model mode view and should be looking at the following screen:

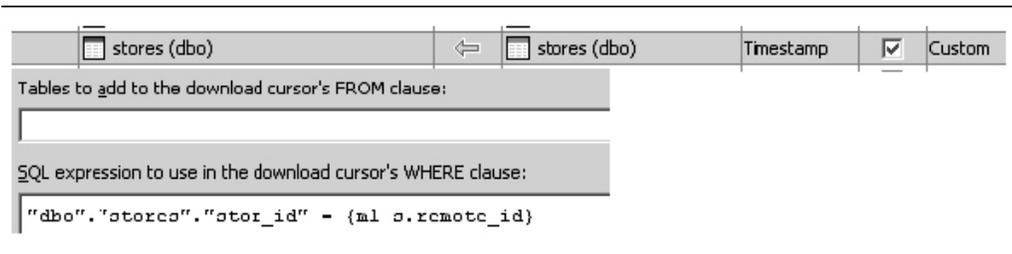
Status	Remote Table ▲	Dir.	Consolidated Table	DrId. Type	Del.	DrId. Sub.
	au_px (dbo)	←	au_px (dbo)	Timestamp	<input checked="" type="checkbox"/>	None
	authors (dbo)	←	authors (dbo)	Timestamp	<input checked="" type="checkbox"/>	None
	sales (dbo)	↔	sales (dbo)	Timestamp	<input checked="" type="checkbox"/>	None
	salesdetail (dbo)	↔	salesdetail (dbo)	Timestamp	<input checked="" type="checkbox"/>	None
	stores (dbo)	←	stores (dbo)	Timestamp	<input checked="" type="checkbox"/>	None
	titleauthor (dbo)	←	titleauthor (dbo)	Timestamp	<input checked="" type="checkbox"/>	None
	titles (dbo)	←	titles (dbo)	Timestamp	<input checked="" type="checkbox"/>	None

17. The Dir. column specifies the direction the data is passed (download-only, upload-only or bi-directional). Set the directions are follows:

- SALES and SALESDETAIL are bi-directional
- The remaining tables are download only

18. Set up the row filtering procedure that you skipped in step 13. In a MobiLink environment, each remote database is given a remote ID. You can use this remote ID to filter rows. You do this in the download\_cursor script for each synchronized table. The download cursor script specifies what columns and rows are downloaded from each table to the remote database. To filter rows by remote ID, you add a restriction to the WHERE clause of the download cursor.

- For the STORES table, change the value in the Download Subset column to Custom.
- Click the Download Subset tab at the bottom of the screen.
- Fill in the text boxes on this tab as follows:



This ensures that you only download information about one store, namely, the store that has an identifier that equals the remote ID for the database. Later you must set up the remote database with a remote ID equal to a particular store's identifier.

d. Follow these steps for the SALES, SALESDETAIL, and DISCOUNTS tables.

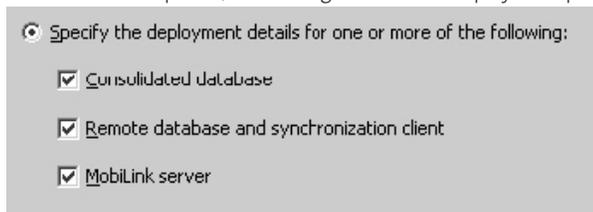
Make sure you rename the table in the expression appropriately.

19. Save the synchronization model.

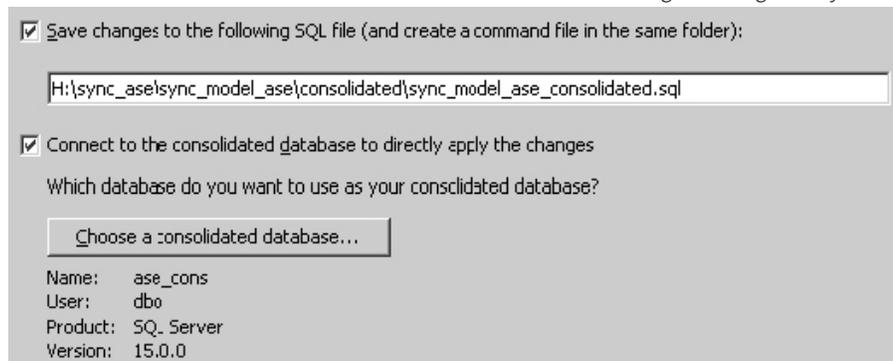
The synchronization model is complete and ready to be deployed.

### DEPLOYING THE SYNCHRONIZATION MODEL

1. In the left pane, right-click the synchronization model you just created and choose Deploy from the popup menu.
2. Select all of the options, instructing MobiLink to deploy each piece of the synchronization environment.



3. MobiLink generates a .sql file with the commands it needs to execute on the consolidated database. Specify the location of the .sql file, and choose to connect to the consolidated database and have MobiLink run the file against it right away.



4. Choose New SQL Anywhere Database to specify the type of remote database being deployed.

**New SQL Anywhere database**

This will create a new, full-featured database for Windows (including Windows CE) or UNIX platforms, using the remote schema from the synchronization model and default database creation options.

**New UltraLite database**

This will create a new, compact database for Windows CE, Windows XP, Palm OS or Symbian OS, using the remote schema from the synchronization model and default database creation options.

**Existing SQL Anywhere or UltraLite database**

Choose this option if you already have a SQL Anywhere or UltraLite remote database. If the database has no tables, then deployment will add the remote schema from the synchronization model.

5. MobiLink generates another .sql file with the commands to set up the remote database with all schema and synchronization information. The wizard can also create a new database file right now and initialize it. If you choose not to do this, you must generate a new remote database, and then run the .sql file against it. Select both checkboxes, and specify locations for the SQL file and the remote database file.

**Make a command file and a SQL file with commands to create the database**

This will create files that you can run later to create a remote database. You must specify where you want to save the SQL file. The command file will be created in the same folder.

SQL file:

**Create a remote SQL Anywhere database**

This will create a remote SQL Anywhere database with default database creation options. You must specify where you want to save the main database file. A transaction log file will also be created in the same folder.

SQL Anywhere database file:

6. Specify a MobiLink user name and password, and then click Next. The MobiLink user is stored in the MobiLink server and is used to authenticate the user during synchronization.

What user name do you want to use for connecting to the MobiLink server?

What password do you want to use?

Note: The password is not masked above because it will be stored as plain text in generated command files (and the remote database if you are using a SQL Anywhere remote database). It can be empty if you do not want to use password protection with default MobiLink authentication.

**Register this user in the consolidated database for MobiLink authentication**

7. Specify the communication protocol the MobiLink server uses.
- Choose TCP/IP as the communication protocol
  - Enter 2439 as the port to use

Which type of communication stream and port number do you want to use?

**TCP/IP**

Use TCP/IP for the best performance.

- c. Click Next.

8. Specify what computer your MobiLink server will reside on.

You can type localhost, your computer's name or IP address, or the name or IP address of another network server you want to use.

What is the host name or IP address of the MobiLink server computer?

Host:

If you want to specify additional optional stream parameters for the MobiLink client, type them below as a semicolon-separated list.

9. You can leave the screen blank that asks you to specify additional connection parameters.

If you want to specify additional optional stream parameters for the MobiLink server, type them below as a semicolon-separated list.

Additional server stream options:

10. Set the log verbosity for the MobiLink server and specify the location of the log file. You can use this log for troubleshooting if you encounter any errors.

- a. Choose the amount of log verbosity.
- b. Type a location for the log file.
- c. Click Next.

How much information do you want in the server window and message log?

No information per synchronization except warnings and errors

Minimum per-synchronization information (-v)

Typical debug verbosity (-vcnrs)

Maximal verbosity (-v+)

Note: A high level of verbosity can adversely affect performance and should only be used during development.

Save message log to the following file (using -ot switch):

11. Type the name for the MobiLink server, and specify the location of the command file that you can use to start the MobiLink server.

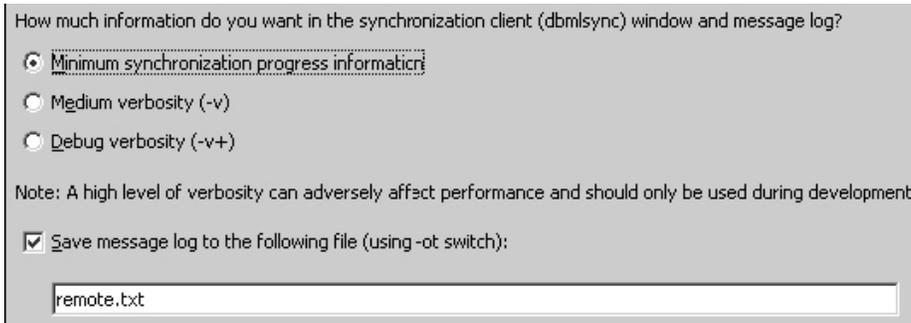
What name do you want to give the MobiLink server?

If you want to specify additional command-line parameters for the MobiLink server, type them below.

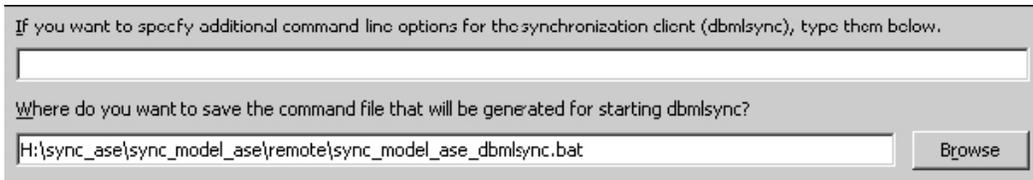
Additional MobiLink server command-line options:

Where do you want to save the command file that will be generated for starting the MobiLink server?

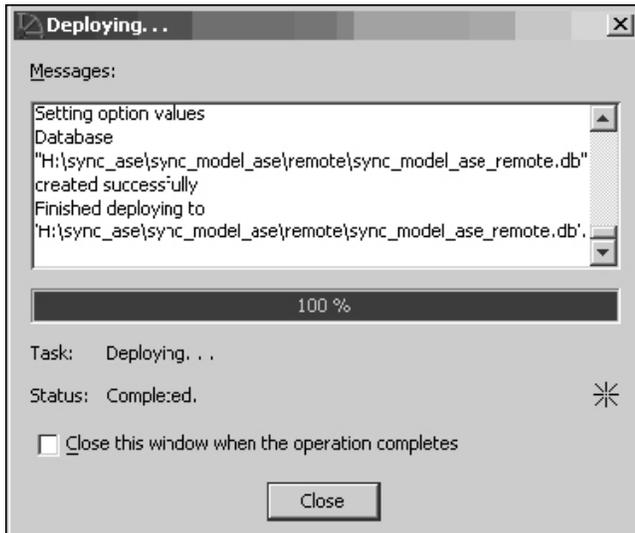
12. Specify the verbosity of the remote synchronization client and choose the location of the remote database log file.



13. Specify advanced options pertaining to the remote client. MobiLink generates a command file that you can use to start synchronization.
  - a. Specify the name and location of the command file.
  - b. Click Next.



14. If your deployment is successful you see the following screen.



- a. Click Close.

You are finished with the deployment wizard. All your files are initialized and can be found in the folder you specified earlier. Your consolidated database is fully configured for synchronization with many remote clients, and you have successfully deployed one remote client. If you want to deploy other remote clients, you can run this wizard again, making sure to create a new MobiLink user and opt out of deploying the consolidated database and MobiLink server. Since these have already been deployed, all you need to do is deploy other remote synchronization clients.

## SYNCHRONIZING

### Starting the consolidated database

1. Ensure that your Adaptive Server Enterprise 15 service is running.  
The MobiLink server must interact with the consolidated database (pubs2) to synchronize the remote clients.

### Starting the MobiLink server

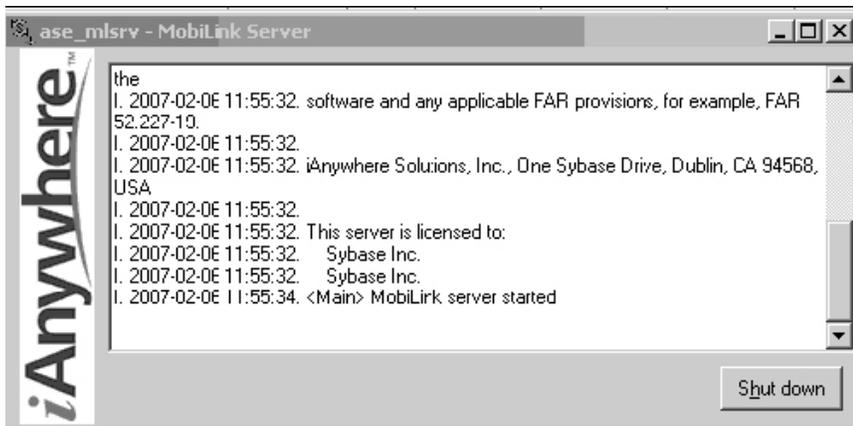
1. At a command prompt, navigate to the folder where you created the synchronization model. (This is the root directory you chose in the first step of the synchronization model wizard.) If you used the suggested directory names, you should have the following directories located in the root directory: `sync_model_ase\mlsrv`.

2. Run the following command from the mlsrv directory:

```
sync_model_ase_mlsrv.bat "dsn=ase_cons;uid=your-aselogin;pwd=your-ase-pwd;"
```

- `sync_model_ase_mlsrv.bat` is the command file created to start the MobiLink server.
- `dsn` is your ODBC data source name.
- `uid` is the user name you use to connect to the consolidated database (the default for Adaptive Server Enterprise is sa).
- `pwd` is the password you use to connect to the consolidated database (as defined when you installed Adaptive Server Enterprise).

If this command runs successfully, the MobiLink server starts and a window similar to the following appears:



If the MobiLink server fails to start, check your consolidated database connection information.

### Starting the remote database

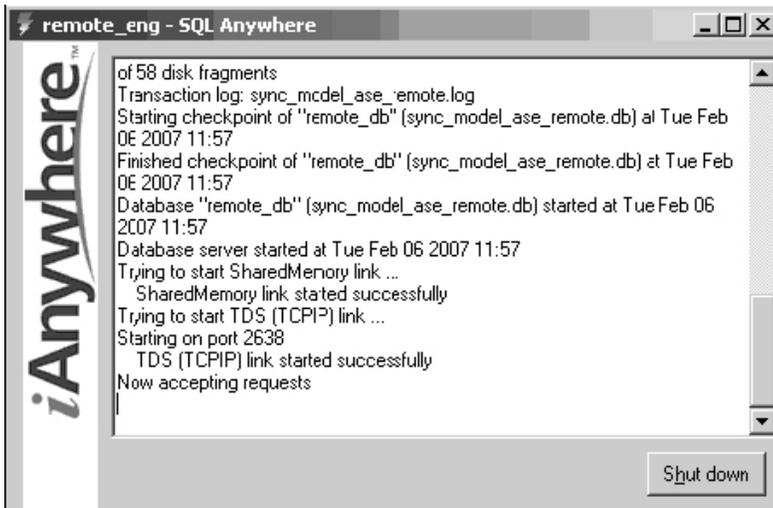
1. At a command prompt, navigate to the directory where the deployment wizard created your remote database.  
If you used the suggested directory names, you should have the following directories located in the root directory: `sync_model_ase\remote`.

2. Start your remote SQL Anywhere database with the following command:

```
dbeng10 -n remote_eng sync_model_ase_remote.db -n remote_db
```

- `dbeng10` is the database server used to start the SQL Anywhere database
- `remote_eng` is the database server name.
- `sync_model_ase_remote.db` is the database file that is started on `remote_eng`.
- `remote_db` is the name of the database on `remote_eng`.

If this command runs successfully, a SQL Anywhere database server named `remote_eng` starts and loads the database called `remote_db`. You should see a window that is similar to the following:



### Initializing the remote ID

In the remote schema, each remote database represents one store. The synchronization scripts you wrote included logic that instructed the MobiLink server to download a subset of data based on the remote ID of the remote database. You must set the database's remote ID to the value of a valid store identifier.

It is important to complete this step before the first synchronization because when the remote device synchronizes for the first time, it downloads all information related to the store (in this case, Thoreau Reading Discount Chain).

1. Choose a valid store identifier.

For a list of valid store identifiers, use the Adaptive Server Enterprise 15 Interactive SQL application to view the STORES table by executing the statement `SELECT * FROM STORES`. For example:

```
select * from stores
```

Results			
	stor_id	stor_name	stor_address
1	5023	Thoreau Reading Discount Chain	20435 Walden Expressway
2	6380	Eric the Read Books	788 Catamaugus Ave.
3	7066	Barnum's	567 Pasadena Ave.
4	7067	News & Brews	577 First St.
5	7131	Doc-U-Mat: Quality Laundry and Books	24-A Avrogado Way
6	7896	Fricative Bookshop	89 Madison St.
7	8042	Bookbeat	679 Carson St.

In this example, the remote database will represent the Thoreau Reading Discount Chain store. Therefore, you must set the database's remote ID to 5023.

2. At a command prompt, execute the following command:

```
dbisql -c "eng=remote_eng;dbn=remote_db;uid=DBA;pwd=sql" "SET OPTION PUBLIC.ml_remote_id='5023'"
```

- **dbisql** is the application used to execute SQL commands against a SQL Anywhere database.
- **eng** sets the database server name to remote\_eng.
- **dbn** sets the database name to remote\_db.
- **uid** sets the user name DBA used to connect to your remote database.
- **pwd** sets the password sql used to connect to your remote database.
- **SET OPTION PUBLIC.ml\_remote\_id='5023'** is the SQL command used to set the remote ID to a value of 5023.

## Synchronizing the remote client

Now you are ready to synchronize the remote client for the first time. This is done with the MobiLink client program `dbmlsync`. `Dbmlsync` connects to the remote database, authenticates itself with the MobiLink server, and performs all of the uploads and downloads necessary to synchronize the remote and consolidated databases based on a publication in the remote database.

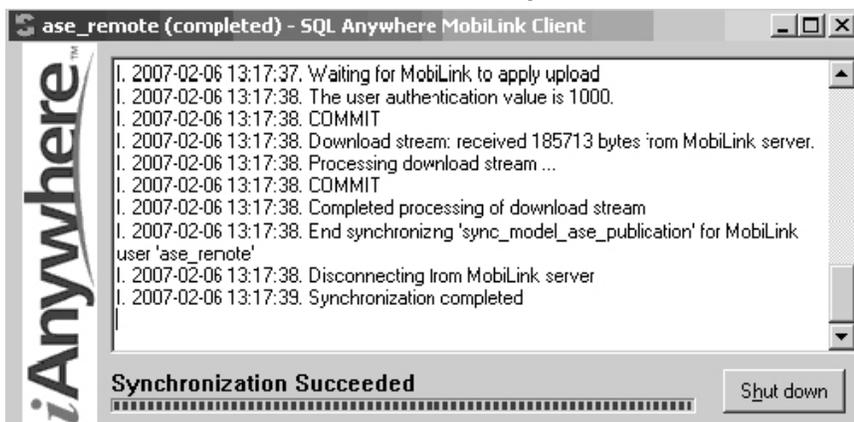
1. At a command prompt, execute the following command:

```
dbmlsync -c "eng=remote_eng;dbn=remote_db;uid=DBA;pwd=sql;" -n sync_model_ase_publication -u ase_remote -mp ase_pass
```

- `dbmlsync` is the synchronization application.
- `eng=remote_eng` specifies the name of the remote database server.
- `dbn=remote_db` specifies the name of the remote database.
- `uid` is the user name used to connect to the remote database.
- `pwd` is the password used to connect to the remote database.
- `sync_model_ase_publication` is the publication on the remote device that will be used to perform the synchronization.
- `ase_remote` is the user name used to authenticate with the MobiLink server.
- `ase_pass` is the password used to authenticate with the MobiLink server.

If you are running the `dbmlsync` application on a different computer from your MobiLink server, you must pass in arguments that specify the location of the MobiLink server.

2. If this command runs successfully, the `dbmlsync` application populates the remote database with a subset of information from the consolidated database. You should see a screen similar to the following:



If synchronization fails, you can start by checking the connection information you pass to the `dbmlsync` application, as well as the MobiLink user name and password. Failing that, check the publication name you used, and ensure the consolidated database and MobiLink server are running. You can also examine the contents of the synchronization logs (server and client).

## Viewing remote client data in Sybase Central

After successfully synchronizing the remote client to the consolidated database through the MobiLink server, the remote data should be populated with information relevant to one store. You can verify this in Sybase Central using the SQL Anywhere 10 plug-in.

1. Open Sybase Central and use SQL Anywhere 10 to make a new connection.

If you haven't stopped the databases that were started in the above sections, your connection information should look like this:

- **User name** DBA
- **Password** sql
- **server name** remote\_eng
- **database name** remote\_db

Once you are connected, you can navigate to the remote tables to view your data.

2. Choose any table and then click the Data tab. The sales table should look like the following:

sales (dbo)			
Columns	Constraints	Referencing Constraints	Indexes
stor_id	ord_num	date	
1 5023	AB-123-DEF-425-1Z3	1985-10-31 00:00:00.000	
2 5023	AB-872-DEF-732-2Z1	1985-11-06 00:00:00.000	
3 5023	AX-532-FED-452-2Z7	1990-12-01 00:00:00.000	
4 5023	BS-345-DSE-860-1F2	1986-12-12 00:00:00.000	
5 5023	GH-542-NAD-713-9F9	1987-03-15 00:00:00.000	
6 5023	NF-123-ADS-642-9G3	1987-07-18 00:00:00.000	
7 5023	XS-135-DER-432-8J2	1991-03-21 00:00:00.000	
8 5023	ZA-000-ASD-324-4D1	1988-07-27 00:00:00.000	
9 5023	ZD-123-DFG-752-9G8	1991-03-21 00:00:00.000	
10 5023	ZS-645-CAT-415-1B2	1991-03-21 00:00:00.000	
11 5023	ZZ-999-ZZZ-999-0A0	1991-03-21 00:00:00.000	

Notice that all the records in this table are for a store with an identifier of 5023. This particular store is not concerned with the sales information of other stores. For this reason, you set the synchronization scripts to filter out rows by the remote ID, and you set this database's remote ID to the value of a particular store identifier. Now this particular store's database takes up less space, and requires less time to synchronize. Since the remote database size is kept to a minimum, frequently performed operations such as entering a new sale or processing a refund on a previous sale can run to completion faster and more efficiently.

## CONCLUSION

Upon completion of this demonstration, you have successfully mobilized an Adaptive Server Enterprise database. It is now set up with one remote synchronization client, and is capable of deploying many more with little time or effort.

To summarize, you completed the following tasks to configure the synchronization environment:

- Designed a remote schema based on a subset of the consolidated schema. You did this by choosing each remote to represent a particular store. The remote database was given a remote ID equal to that store's store identifier.
- Prepared the consolidated database.
- Ensured that each synchronized table had a primary key.
- Created an ODBC data source. The ODBC data source pointed to the Adaptive Server Enterprise 15 database so that MobiLink could access it.
- Created a synchronization model.
- Defined how data was uploaded and downloaded to and from the consolidated database.
- Deployed the synchronization model.
- Made the necessary modifications to the consolidated database.
- Set up the MobiLink server.
- Created a remote database.
- Generated command files to allow for simple initialization.
- Started the remote database.
- Started a new SQL Anywhere database server running the remote database.
- Set the remote database's remote ID.
- Synchronized the remote database to the consolidated database.
- Viewed data in the remote database.
- Verified that data was in fact a subset of the consolidated database, all pertinent to the particular store whose store identifier you used.

## LEGAL NOTICE

Copyright © 2007 iAnywhere Solutions, Inc. All rights reserved. Sybase, the Sybase logo, iAnywhere Solutions, the iAnywhere Solutions logo, Adaptive Server, Mobilink, and SQL Anywhere are trademarks of Sybase, Inc. or its subsidiaries. All other trademarks are property of their respective owners.

The information, advice, recommendations, software, documentation, data, services, logos, trademarks, artwork, text, pictures, and other materials (collectively, "Materials") contained in this document are owned by Sybase, Inc. and/or its suppliers and are protected by copyright and trademark laws and international treaties. Any such Materials may also be the subject of other intellectual property rights of Sybase and/or its suppliers all of which rights are reserved by Sybase and its suppliers.

Nothing in the Materials shall be construed as conferring any license in any Sybase intellectual property or modifying any existing license agreement.

The Materials are provided "AS IS", without warranties of any kind. SYBASE EXPRESSLY DISCLAIMS ALL REPRESENTATIONS AND WARRANTIES RELATING TO THE MATERIALS, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. Sybase makes no warranty, representation, or guaranty as to the content, sequence, accuracy, timeliness, or completeness of the Materials or that the Materials may be relied upon for any reason.

Sybase makes no warranty, representation or guaranty that the Materials will be uninterrupted or error free or that any defects can be corrected. For purposes of this section, 'Sybase' shall include Sybase, Inc., and its divisions, subsidiaries, successors, parent companies, and their employees, partners, principals, agents and representatives, and any third-party providers or sources of Materials.

## Contact Us

**Sybase iAnywhere Worldwide Headquarters** One Sybase Drive, Dublin, CA, 94568 USA

**Phone** 1-800-801-2069 (in US and Canada)

**Fax** 1-519-747-4971

**World Wide Web** <http://www.iAnywhere.com>

**Email** [contact.us@iAnywhere.com](mailto:contact.us@iAnywhere.com)