iAnywhere™ SOLUTIONS
A SYBASE COMPANY

Introducing SQL Anywhere® Studio

Part number: 38120-01-0900-01

Last modified: June 2003

Contents

About This Manual

Subject This book introduces SQL Anywhere Studio, the complete relational data-management and synchronization system for mobile, embedded, and workgroup computing.

SQL Anywhere Studio is a set of software components for working with relational data. It includes the Adaptive Server Anywhere and UltraLite relational database-management systems, as well as data synchronization and replication technology. It also includes applications for designing and deploying databases, and for creating custom reports and data entry forms.

This book uses hands-on tutorials to introduce you to the components of SQL Anywhere Studio, showing you how they fit together to build everything from single-user database applications to distributed computing systems synchronizing data across thousands of databases.

Audience This book is for all application developers and database administrators using SQL Anywhere Studio.

Before you begin This book assumes an elementary familiarity with relational databases and SQL. If you do not have such a familiarity, you should read *Adaptive Server Anywhere Getting Started.*

SQL Anywhere Studio documentation

This book is part of the SQL Anywhere documentation set. This section describes the books in the documentation set and how you can use them.

The SQL Anywhere Studio documentation

The SQL Anywhere Studio documentation is available in a variety of forms: in an online form that combines all books in one large help file; as separate PDF files for each book; and as printed books that you can purchase. The documentation consists of the following books:

- **Introducing SQL Anywhere Studio** This book provides an overview of the SQL Anywhere Studio database management and synchronization technologies. It includes tutorials to introduce you to each of the pieces that make up SQL Anywhere Studio.

- **What's New in SQL Anywhere Studio** This book is for users of previous versions of the software. It lists new features in this and previous releases of the product and describes upgrade procedures.

- **Adaptive Server Anywhere Getting Started** This book is for people new to relational databases or new to Adaptive Server Anywhere. It provides a quick start to using the Adaptive Server Anywhere database-management system and introductory material on designing, building, and working with databases.

- **Adaptive Server Anywhere Database Administration Guide** This book covers material related to running, managing, and configuring databases and database servers.

- **Adaptive Server Anywhere SQL User's Guide** This book describes how to design and create databases; how to import, export, and modify data; how to retrieve data; and how to build stored procedures and triggers.

- **Adaptive Server Anywhere SQL Reference Manual** This book provides a complete reference for the SQL language used by Adaptive Server Anywhere. It also describes the Adaptive Server Anywhere system tables and procedures.

- **Adaptive Server Anywhere Programming Guide** This book describes how to build and deploy database applications using the C, C++, and Java programming languages. Users of tools such as Visual Basic and PowerBuilder can use the programming interfaces provided by those tools. It also describes the Adaptive Server Anywhere ADO.NET data provider.

- **Adaptive Server Anywhere Error Messages** This book provides a complete listing of Adaptive Server Anywhere error messages together with diagnostic information.

- **SQL Anywhere Studio Security Guide** This book provides information about security features in Adaptive Server Anywhere databases. Adaptive Server Anywhere 7.0 was awarded a TCSEC (Trusted Computer System Evaluation Criteria) C2 security rating from the U.S. Government. This book may be of interest to those who wish to run the current version of Adaptive Server Anywhere in a manner equivalent to the C2-certified environment.

- **MobiLink Synchronization User's Guide** This book describes how to use the MobiLink data synchronization system for mobile computing, which enables sharing of data between a single Oracle, Sybase, Microsoft or IBM database and many Adaptive Server Anywhere or UltraLite databases.

- **MobiLink Synchronization Reference** This book is a reference guide to MobiLink command line options, synchronization scripts, SQL statements, stored procedures, utilities, system tables, and error messages.

- **iAnywhere Solutions ODBC Drivers** This book describes how to set up ODBC drivers to access consolidated databases other than Adaptive Server Anywhere from the MobiLink synchronization server and from Adaptive Server Anywhere remote data access.

- **SQL Remote User's Guide** This book describes all aspects of the SQL Remote data replication system for mobile computing, which enables sharing of data between a single Adaptive Server Anywhere or Adaptive Server Enterprise database and many Adaptive Server Anywhere databases using an indirect link such as e-mail or file transfer.

- **SQL Anywhere Studio Help** This book includes the context-sensitive help for Sybase Central, Interactive SQL, and other graphical tools. It is not included in the printed documentation set.

- **UltraLite Database User's Guide** This book is intended for all UltraLite developers. It introduces the UltraLite database system and provides information common to all UltraLite programming interfaces.

- **UltraLite Interface Guides** A separate book is provided for each UltraLite programming interface. Some of these interfaces are provided as UltraLite components for rapid application development, and others are provided as static interfaces for C, C++, and Java development.

In addition to this documentation set, PowerDesigner and InfoMaker include their own online documentation.

Documentation formats

SQL Anywhere Studio provides documentation in the following formats:

- **Online documentation** The online documentation contains the complete SQL Anywhere Studio documentation, including both the books and the context-sensitive help for SQL Anywhere tools. The online documentation is updated with each maintenance release of the product, and is the most complete and up-to-date source of documentation.

  To access the online documentation on Windows operating systems, choose Start ➤ Programs ➤ SQL Anywhere 9 ➤ Online Books. You can navigate the online documentation using the HTML Help table of contents, index, and search facility in the left pane, as well as using the links and menus in the right pane.

  To access the online documentation on UNIX operating systems, see the HTML documentation under your SQL Anywhere installation.

- **Printable books** The SQL Anywhere books are provided as a set of PDF files, viewable with Adobe Acrobat Reader.

  The PDF files are available on the CD ROM in the *pdf_docs* directory. You can choose to install them when running the setup program.

- **Printed books** The complete set of books is available from Sybase sales or from eShop, the Sybase online store. You can access eShop by clicking How to Buy ➤ eShop at *http://www.ianywhere.com*.

Documentation conventions

This section lists the typographic and graphical conventions used in this documentation.

Syntax conventions

The following conventions are used in the SQL syntax descriptions:

- **Keywords** All SQL keywords appear in upper case, like the words ALTER TABLE in the following example:

  ALTER TABLE [*owner*.]*table-name*

- **Placeholders** Items that must be replaced with appropriate identifiers or expressions are shown like the words *owner* and *table-name* in the following example:

  ALTER TABLE [*owner*.]*table-name*

- **Repeating items** Lists of repeating items are shown with an element of the list followed by an ellipsis (three dots), like *column-constraint* in the following example:

  ADD *column-definition* [*column-constraint*, . . .]

  One or more list elements are allowed. In this example, if more than one is specified, they must be separated by commas.

- **Optional portions** Optional portions of a statement are enclosed by square brackets.

  RELEASE SAVEPOINT [*savepoint-name*]

  These square brackets indicate that the *savepoint-name* is optional. The square brackets should not be typed.

- **Options** When none or only one of a list of items can be chosen, vertical bars separate the items and the list is enclosed in square brackets.

  [**ASC** | **DESC**]

  For example, you can choose one of ASC, DESC, or neither. The square brackets should not be typed.

- **Alternatives** When precisely one of the options must be chosen, the alternatives are enclosed in curly braces and a bar is used to separate the options.

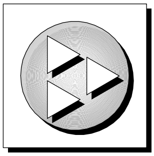
  [**QUOTES** { **ON** | **OFF** }]

  If the QUOTES option is used, one of ON or OFF must be provided. The brackets and braces should not be typed.

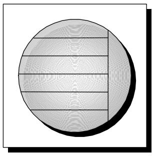
Graphic icons

The following icons are used in this documentation.

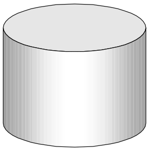
- A client application.



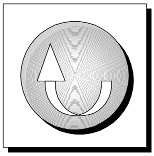
- A database server, such as Sybase Adaptive Server Anywhere.



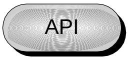
- A database. In some high-level diagrams, the icon may be used to represent both the database and the database server that manages it.



- Replication or synchronization middleware. These assist in sharing data among databases. Examples are the MobiLink Synchronization Server and the SQL Remote Message Agent.



- A programming interface.



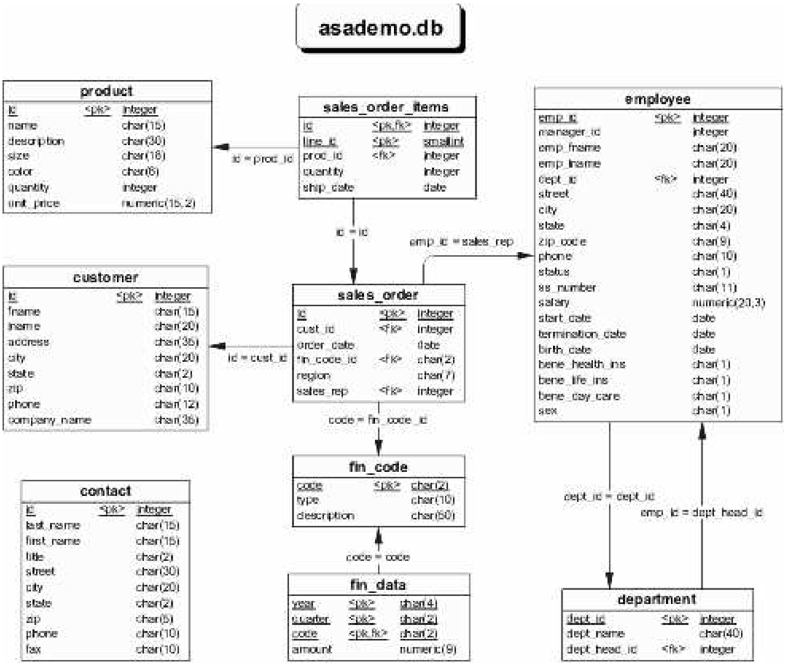
The Adaptive Server Anywhere sample database

Many of the examples throughout the documentation use the Adaptive Server Anywhere sample database.

The sample database is held in a file named *asademo.db*, and is located in your SQL Anywhere directory.

The sample database represents a small company. It contains internal information about the company (employees, departments, and finances) as well as product information and sales information (sales orders, customers, and contacts). All information in the database is fictional.

The following figure shows the tables in the sample database and how they relate to each other.



product

id	<pk>	integer
name		char(15)
description		char(30)
size		char(18)
color		char(6)
quantity		integer
unit_price		numeric(15,2)

sales_order_items

id	<pk,fk>	integer
line_id	<pk>	smallint
prod_id	<fk>	integer
quantity		integer
ship_date		date

employee

emp_id	<pk>	integer
manager_id		integer
emp_fname		char(20)
emp_lname		char(20)
dept_id	<fk>	integer
street		char(40)
city		char(20)
state		char(4)
zip_code		char(9)
phone		char(10)
status		char(1)
ss_number		char(11)
salary		numeric(20,3)
start_date		date
termination_date		date
birth_date		date
bene_health_ins		char(1)
bene_life_ins		char(1)
bene_day_care		char(1)
sex		char(1)

customer

id	<pk>	integer
fname		char(15)
lname		char(20)
address		char(35)
city		char(20)
state		char(2)
zip		char(10)
phone		char(12)
company_name		char(35)

sales_order

id	<pk>	integer
cust_id	<fk>	integer
order_date		date
fin_code_id	<fk>	char(2)
region		char(7)
sales_rep	<fk>	integer

fin_code

code	<pk>	char(2)
type		char(10)
description		char(50)

contact

id	<pk>	integer
last_name		char(15)
first_name		char(15)
title		char(2)
street		char(30)
city		char(20)
state		char(2)
zip		char(5)
phone		char(10)
fax		char(10)

fin_data

year	<pk>	char(4)
quarter	<pk>	char(2)
code	<pk,fk>	char(2)
amount		numeric(9)

department

dept_id	<pk>	integer
dept_name		char(40)
dept_head_id	<fk>	integer

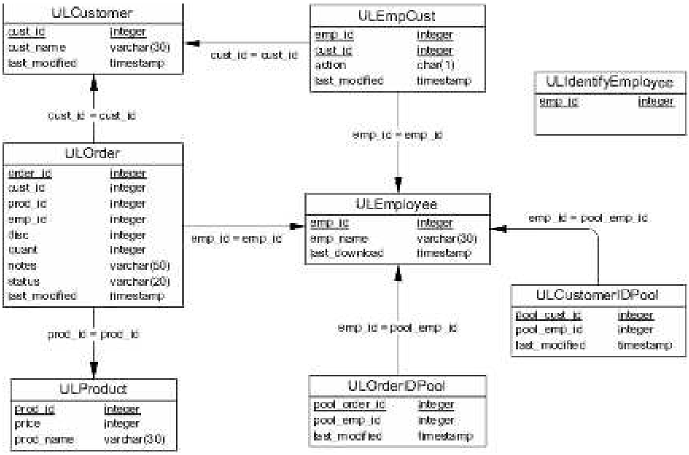
The CustDB sample database

Many of the examples in the MobiLink and UltraLite documentation use the UltraLite sample database.

The reference database for the UltraLite sample database is held in a file named *custdb.db*, and is located in the *Samples\UltraLite\CustDB* subdirectory of your SQL Anywhere directory. A complete application built on this database is also supplied.

The sample database is a sales-status database for a hardware supplier. It holds customer, product, and sales force information for the supplier.

The following figure shows the tables in the CustDB database and how they are related to each other.



Finding out more and providing feedback

We would like to receive your opinions, suggestions, and feedback on this documentation.

You can provide feedback on this documentation and on the software through newsgroups set up to discuss SQL Anywhere technologies. These newsgroups can be found on the *forums.sybase.com* news server.

The newsgroups include the following:

- sybase.public.sqlanywhere.general.
- sybase.public.sqlanywhere.linux.
- sybase.public.sqlanywhere.mobilink.
- sybase.public.sqlanywhere.product_futures_discussion.
- sybase.public.sqlanywhere.replication.
- sybase.public.sqlanywhere.ultralite.

Newsgroup disclaimer

iAnywhere Solutions has no obligation to provide solutions, information or ideas on its newsgroups, nor is iAnywhere Solutions obliged to provide anything other than a systems operator to monitor the service and insure its operation and availability.

iAnywhere Solutions Technical Advisors as well as other staff assist on the newsgroup service when they have time available. They offer their help on a volunteer basis and may not be available on a regular basis to provide solutions and information. Their ability to help is based on their workload.

PART I

SQL ANYWHERE STUDIO OVERVIEW

This part introduces SQL Anywhere Studio and its database management and replication technologies.

CHAPTER 1

Introducing SQL Anywhere Studio

About this chapter

This chapter introduces the pieces that make up SQL Anywhere Studio. It also describes how to install it, and how to use the documentation.

Contents

Welcome to SQL Anywhere Studio

With SQL Anywhere Studio, you can deliver information to workgroup, mobile, and embedded database systems throughout an entire organization.

SQL Anywhere Studio includes the following components. More information on each component can be found later in this book.

Relational database systems

- **Adaptive Server Anywhere** The relational database at the core of the product is a transaction-based SQL database designed for personal and workgroup use. Adaptive Server Anywhere runs on a wide range of operating systems, including many flavors of Windows and UNIX, as well as on Novell NetWare. It runs on hardware ranging from multiple-CPU workgroup servers to the most modest PCs, as well as on Windows CE devices.
- **UltraLite** For building and deploying relational database applications on small devices, including the Palm Computing Platform and Windows CE. UltraLite has built-in support for MobiLink synchronization.

  UltraLite lets you build relational databases with less than 50 kb of disk space, and is specifically intended for small devices.

Data synchronization technologies

- **MobiLink** For two-way synchronization of data between a central database and many remote UltraLite or Adaptive Server Anywhere databases. The central database can be Adaptive Server Anywhere, Adaptive Server Enterprise, or another DBMS such as Oracle, Microsoft SQL Server, or IBM DB2.
- **SQL Remote** For two-way, message-based replication of data between a central database and many remote databases. With SQL Remote, you can replicate data between laptop computers and a central database, using e-mail or dial-up access.
- **Replication Agent** For replicating data from Adaptive Server Anywhere databases to other databases via Sybase Replication Server.

Development, design, and administration tools

- **InfoMaker** For querying databases and creating sophisticated and effective custom reports of data. InfoMaker is also a personal data assistant that lets you work with data in many ways.
- **PowerDesigner** For designing, generating, documenting, and maintaining databases.
- **Management and development tools** SQL Anywhere Studio includes the Sybase Central database management tool and the Interactive SQL database utility, as well as a query editor and a stored procedure debugger.

SQL Anywhere Studio includes an optionally-installable accessibility enablement module. This component provides the Sun AccessBridge module, which is loaded whenever you use Sybase Central or Interactive SQL. Third-party software such as screen readers make use of this module to provide access to software features.

InfoMaker and PowerDesigner are available only on Windows operating systems. However, they can be used as clients of a server running on any supported operating system.

Separately-licensable components

The following components are separately licensable and must be ordered before you can install them. To order these components, see the card in your SQL Anywhere Studio package or see *http://www.sybase.com/detail?id=1015780.*

- **Java option** The Java virtual machine and runtime classes that enable the use of Java in the database must be ordered separately.
- **Security option** The software for data encryption (transport-layer security) between an Adaptive Server Anywhere or MobiLink server and a client, as well as MDSR database file encryption must be ordered separately. AES database file encryption is included in the base package.

Installing SQL Anywhere Studio

How you install SQL Anywhere Studio depends on the operating system you are using. You must ensure that you are properly licensed before installing the software.

❖ To install SQL Anywhere Studio (Windows operating systems)

1. Start the installation by running the *setup.exe* program in the root directory of the SQL Anywhere Studio CD-ROM. Follow the instructions in the setup wizard.

   The setup program allows you to choose which of the components you wish to install.

❖ To install SQL Anywhere Studio (Novell NetWare or Windows CE)

1. You must install Adaptive Server Anywhere for NetWare from a machine connected to the NetWare server and running a Microsoft Windows operating system. Run the SQL Anywhere Studio setup program and choose NetWare or Windows CE installation.

❖ To install SQL Anywhere Studio (UNIX)

1. The installation instructions depend on which UNIX operating system you are using. For more information, consult the separate *Read Me First* booklet, which is included in SQL Anywhere Studio for UNIX.

CHAPTER 2

The Architecture of Database Applications

About this chapter

Adaptive Server Anywhere is a relational database system with a multitude of uses, from a network database server hosting many clients to a compact embedded database. The UltraLite deployment technology allows you to use Adaptive Server Anywhere features on even the smallest of devices.

This chapter describes the architecture of database applications that operate with a single database server.

☞ For information about distributed database systems involving many databases sharing data through SQL Anywhere replication and synchronization technologies, see "Consolidated and remote databases" on page 21.

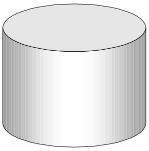
Contents

The pieces of a database system

This section describes how database applications and the database server work together to manage databases.

Any information system contains the following pieces:

- **A database** Data is stored in a database. In diagrams in the documentation, a database is indicated by a cylinder:

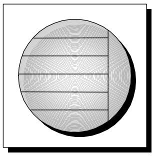
  

  An Adaptive Server Anywhere database is a file, usually with an extension of *.db*. Adaptive Server Anywhere includes a sample database for you to work with: this is the file *asademo.db* in your Adaptive Server Anywhere installation directory.

  UltraLite databases are stored in a device-dependent manner. For example, on Windows CE, the UltraLite database is a file, but on the Palm Computing Platform it is stored in the Palm database.

- **A database server** The database server manages the database. No other applications address the database file directly; they all communicate with the database server.

  In diagrams in the documentation, a database server is indicated as follows:

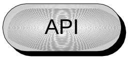
  

  Adaptive Server Anywhere provides two versions of its database server: the personal database server and the network database server. In addition to the features of the personal server, the network server supports client/server communications across a network. The request-processing engine is identical in the two servers.

- **A programming interface** Applications communicate with the database server using a programming interface. You can use ODBC, OLE DB, ADO, JDBC, Sybase Open Client, or embedded SQL.

  The programming interface provides a set of function calls for communicating with the database. For ODBC and JDBC, the library is commonly called a **driver**. The interface is typically provided as a shared library on UNIX operating systems or a dynamic link library (DLL) on PC operating systems. The JDBC interface uses the Sybase jConnect driver, which is a zip file of compiled Java classes.

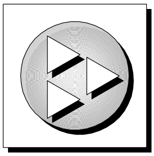
In diagrams in the documentation, a programming interface is indicated as follows:



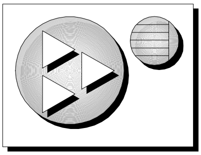
- **A client application** Client applications use one of the programming interfaces to communicate with the database server.

  If you develop an application using a rapid application development (RAD) tool such as Sybase PowerDesigner, PowerJ or PowerBuilder, you may find that the tool provides its own methods for communicating with database servers, and hides the details of the language interface. Nevertheless, all applications do use one of the supported interfaces.

  In diagrams in the documentation, a client application is indicated by the following:

  UltraLite database servers are custom-generated for each UltraLite application, and are part of the application itself. An UltraLite application together with its database server is indicated as follows:

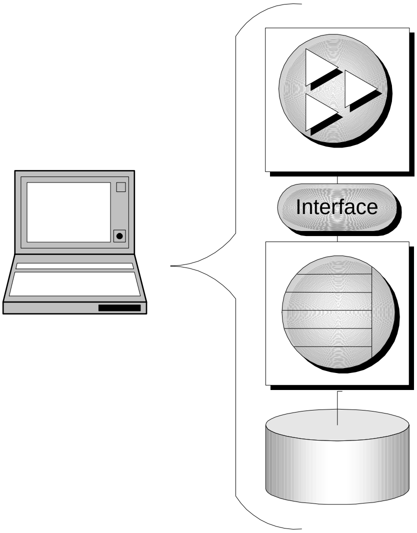
  

Embedded database architecture

You can use Adaptive Server Anywhere to build a complete application and database on a single computer. In the simplest arrangement, this is a **standalone application**: it is self-contained, with no connection to other databases. In this case, it is common to refer to the database as an **embedded database** because, as far as the end user is concerned, the database is a part of the application. When a database server is used as an embedded database, it is sometimes called a **database engine**.

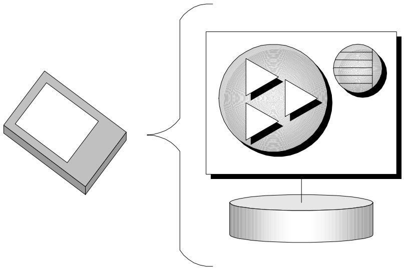
Many relational database management systems require experienced staff for administration. A characteristic of embedded databases is the ability to run entirely without administration.

The Adaptive Server Anywhere personal database server is generally used for standalone applications. A client application connects through a programming interface to a database server running on the same machine:



UltraLite architecture

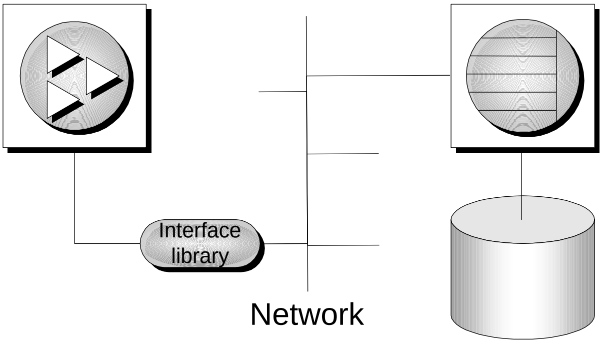
If you want to provide a database application for a small device such as a handheld organizer, you may want to use the UltraLite deployment technology. In UltraLite, the database server and the application are part of the same process, and the database server is specific to the application.



In this case, the database may not be a file on disk. The storage method for the database depends on the deployment platform.

Client/server architecture

You can use Adaptive Server Anywhere to build an installation with many applications, running on different machines, connected over a network to a single database server running on a separate machine. This is a **client/server** environment, and has the following architecture. The interface library is located on each client machine.



In this case the database server is the Adaptive Server Anywhere network database server which supports network communications. The database is also called a **multi-user database**.

No changes are needed to a client application for it to work in a client/server environment, except to identify the server to which it should connect.

Multi-tier computing architecture

In multi-tier computing, application logic is held in an application server, such as Sybase EAServer, which sits between the database server and the client applications. In many situations, a single application server may access multiple databases in addition to non-relational data stores. In the Internet case, client applications are browser-based, and the application server is generally a Web server extension.

Sybase EAServer stores application logic in the form of components, and makes these components available to client applications. The components may be PowerBuilder components, Java beans, or COM components.

Application servers can also provide transaction logic to their client applications—guaranteeing that sets of operations are executed atomically across multiple databases. Adaptive Server Anywhere is well-suited to multi-tier computing, and can participate in distributed transactions coordinated by Microsoft Distributed Transaction Coordinator. Both Sybase Enterprise Application Server and Microsoft Transaction Server use DTC to provide transaction services to their client applications.

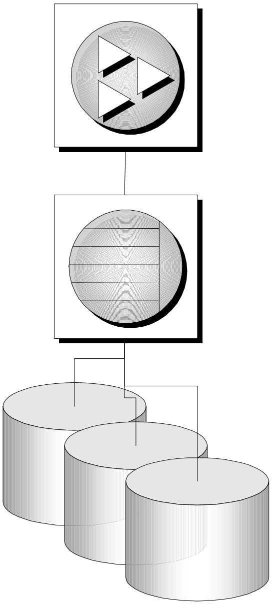
☞ For more information, see “Three-tier Computing and Distributed Transactions” [*ASA Programming Guide,* page 455].

Using multiple databases

This section describes extensions to the Adaptive Server Anywhere architecture described above for the case where you want to use more than one database.

Running multiple databases on a single database server

The Adaptive Server Anywhere personal and network database servers can both manage several databases simultaneously. Each connection from an application must be to a single database, but an application can use separate connections to different databases, or a set of applications can work on different databases, all through the same database server.



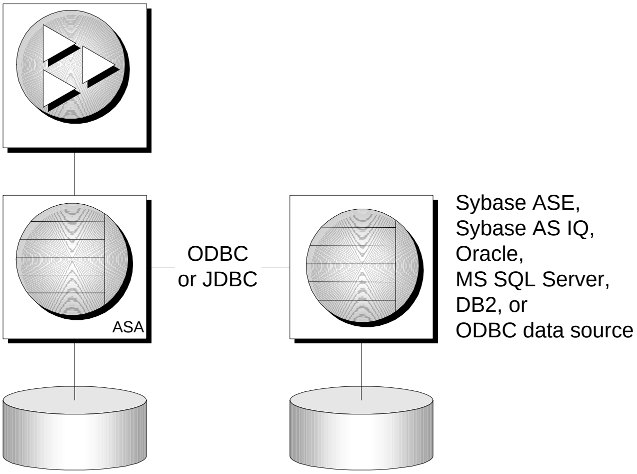
Databases can be started when the database server is started, or by connecting to a database using the DatabaseFile connection parameter.

☞ For more information, see “The database server” [*ASA Database Administration Guide*, page 124], and “DatabaseFile connection parameter [DBF]” [*ASA Database Administration Guide*, page 183].

Accessing data in other databases

You can access databases on multiple database servers, or even on the same server, using the Adaptive Server Anywhere Remote Data Access features.

The application is still connected to a single database as in the architecture diagrams above, but by defining remote servers, you can use proxy tables that exist on the remote database as if they were in the database to which you are connected.



☞ For more information, see “Accessing Remote Data” [*ASA SQL User’s Guide*, page 557].

Comparing Adaptive Server Anywhere and UltraLite

This section highlights the differences between Adaptive Server Anywhere and UltraLite, to help you decide which technology is suited to your needs.

Adaptive Server Anywhere

For many years, Sybase Adaptive Server Anywhere has provided relational database technology designed specifically for the needs of workgroup, mobile and embedded computing. The product has been designed from the ground up with this market in mind:

- Adaptive Server Anywhere is designed to operate efficiently with limited memory, CPU power, and disk space. Core features such as the query optimizer and the data caching mechanism are designed specifically to operate without extravagant use of resources. At the same time, Adaptive Server Anywhere contains the features needed to take advantage of workgroup servers, including support for many users, scalability over multiple CPUs, and advanced concurrency features.

- Adaptive Server Anywhere is a cross-platform solution. The same database runs on Windows (Windows 95 and its successors, Windows NT and its successors, and Windows CE), UNIX including Linux, and Novell NetWare. You can move a database file from one operating system to another.

- Adaptive Server Anywhere is designed to operate without administration, making it ideal for use as an embedded database. Adaptive Server Anywhere provides a self-tuning query optimizer, built-in scheduling and event-handling capabilities, as well as autostart and autostop mechanisms.

- Many years of experience working with successful customers have led to a rich set of field-tested features. Not only the standard checklist features of stored procedures, triggers, declarative referential integrity, full transaction processing, and recovery, but all the little extras that can make the difference between a successful project and a failure.

- SQL Anywhere synchronization technologies (SQL Remote and MobiLink) mean that you can integrate Adaptive Server Anywhere databases into your organization’s infrastructure.

UltraLite: the “small fingerprint” database

The appearance of small computing devices such as handheld computers, pagers, and mobile phones creates a demand for a database with even more modest memory requirements than Adaptive Server Anywhere. An obvious option is to produce a trimmed down relational database engine, but our experience has shown that each application and each customer has a distinct set of features that are, for them, essential. Further, such an approach would mean that customers would have to learn two different databases, inevitably different in some ways.

Instead, we developed UltraLite, a novel technology that uses a reference database and your application source code to generate a relational database engine containing only those features of Adaptive Server Anywhere used by your application. Each query is stored with a complete access plan for fast execution; the code needed to execute just those tasks you need are built into your UltraLite database engine. Each UltraLite database engine is different, but many are only a few tens of kilobytes, and can easily be run in a device as small as a pager.

Adaptive Server Anywhere serves as a reference database when you build your UltraLite application, and so your SQL statements, data types, and so on are exactly those of Adaptive Server Anywhere. UltraLite is a deployment technology for Adaptive Server Anywhere, not a different database system. The tasks that each UltraLite database engine is built to perform are carried out in a manner completely compatible with Adaptive Server Anywhere.

UltraLite has built-in MobiLink synchronization technology so that your application is linked into the information network.

UltraLite supports the Palm Computing Platform, Windows CE, and other operating systems used in small devices, such as Java.

Choosing a database

This section describes the complementary features of Adaptive Server Anywhere and UltraLite databases, to help you choose the most appropriate technology for your needs.

If you are deploying mainly to PCs, Adaptive Server Anywhere is built to meet your needs. If you are deploying to small devices such as the Palm Computing Platform, UltraLite will fit the bill. If you are planning to deploy on a platform supported by both technologies, such as Windows CE, you should consider the following issues.

♦ **Application type** UltraLite database engines include only the code necessary to execute the tasks specified when your application is compiled, so you cannot use dynamic SQL to execute ad hoc queries against an UltraLite database. Further, each UltraLite database and database engine is for use by only a single application: if you want to use more than one application against a single database, you should choose Adaptive Server Anywhere.

- **Programming interface** If you are happier using an interface other than Embedded SQL or Java, such as ODBC or OLE DB, and your target platform is not so resource-constrained as to require UltraLite, you can use Adaptive Server Anywhere.
- **Deploying across a variety of devices** The programming model for UltraLite enables platform-independent database access code so that you can port the user interface to new platforms and devices without having to alter the underlying data access layer.

CHAPTER 3

Replication Technologies

About this chapter

Data **replication** is the sharing of data among physically distinct databases. When an application modifies shared data at any one database, the changes are propagated to the other databases in the replication setup. Changes can be propagated by various means and through a variety of channels, allowing flexible replication setups while preserving data integrity. Data replication is also referred to as data **synchronization**.

Sybase has three replication technologies. **MobiLink** and **SQL Remote** are designed for replication between a central database and a large number of remote databases. **Replication Server** is intended for near-real-time replication between a relatively small number of databases.

Contents

Introduction

This section introduces basic concepts in data replication.

Benefits of data replication

Data availability

One of the key benefits of a data replication system is that data is made available locally, rather than through potentially expensive, less reliable, and slow connections to a single central database. Data is accessible locally even in the absence of any connection to a central server, so you are not cut off from data in the event of a failure of a long-distance network connection.

Response time

Replication improves response times for data requests for two reasons. Retrieval rates are faster because requests are processed on a local server, without accessing a wide area network. Also, local processing offloads work from a central database server so that competition for processor time is decreased.

Challenges for replication technologies

Any replication technology must address several challenges that arise as a result of the increased flexibility permitted by replication.

Transactional integrity

One of the challenges of any replication system is to ensure that each database retains transactional integrity at all times.

Replication Server and SQL Remote replicate portions of the transaction log in such a way that transactions are maintained during replication: either a whole transaction is replicated, or none of it is replicated. This ensures transactional integrity at each database in the setup.

MobiLink consolidates changes made in multiple, committed transactions. These changes are applied to another database in a single transaction.

Data consistency

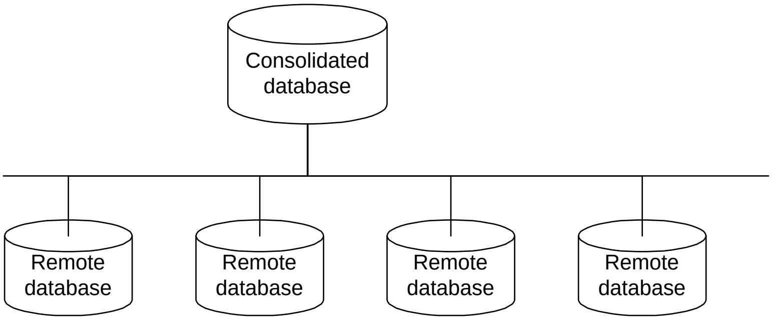
Another challenge to replication systems is to maintain data consistency throughout the setup. Replication systems maintain a **loose consistency** in the setup as a whole: that is, all changes are replicated to each site over time in a consistent manner, but different sites may have different copies of data at any instant.

Consolidated and remote databases

Both MobiLink and SQL Remote provide data replication between a consolidated database and a set of remote databases.

A **consolidated database** is a database that contains all the data to be replicated. A **remote database** is a copy of the consolidated database that may be running either at the same site as the consolidated database or at a physically distant site.

The figure displays a schematic illustration of a small installation.



Remote users

A replication installation includes many remote databases. Each remote database contains a subset of the information in the consolidated database. Each remote database is a physically separate database, usually on a separate computer. All remote databases must stay consistent with the consolidated database.

The entire replication setup may be considered a single dispersed database, with the master copy of all shared data being kept at the consolidated database.

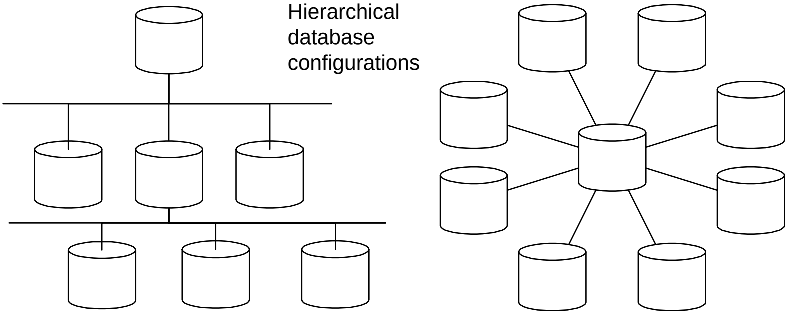
Each remote site that submits replications to the consolidated database is considered to be a **remote user** of the consolidated database. In the case that a remote site is a multi-user server, the entire site is considered to be a single remote user of the consolidated database.

Hierarchical database configurations

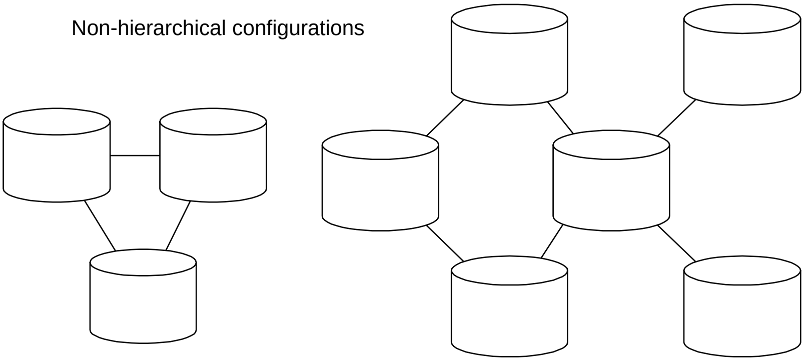
For databases in a **hierarchical configuration**, every database has a single parent database, except the consolidated database, which has no parent.

SQL Remote supports hierarchical configurations of databases; it does not support peer-to-peer replication or other non-hierarchical configurations. MobiLink is also normally used with a hierarchical configuration, but can also be used in other configurations.

For any two databases directly sharing data in a hierarchical configuration, one is always above or below the other in the hierarchy.



Databases in a non-hierarchical configuration do not have a well-defined notion of above or below.



In a MobiLink or SQL Remote installation, each database contains all of, or a subset of, the data replicated by the database above it in the hierarchy.

Remote databases can contain tables that are not present at the consolidated database as long as they are not involved in replication. SQL Remote requires that the table and column names in the remote databases match the ones in the consolidated database. In contrast, MobiLink allows data to be stored in different columns and tables in the remote databases than in the consolidated database, allowing greater flexibility.

Two-way replication

All Sybase replication technologies provide two-way replication: changes made at the consolidated database are propagated to remote databases and changes made at remote databases are propagated to the consolidated database and, hence, to other remote databases. Sybase Replication Server requires that a particular piece of data can be modified at only one location.

Both SQL Remote and MobiLink allow the same data to be changed simultaneously at multiple locations and provide a means of resolving any conflicts.

Propagation methods

When a transaction modifies shared data at any one database, the transaction or changes must be replicated to the other databases in the replication setup. There are various means by which this task may be accomplished.

Session-based replication: MobiLink

In a session-based replication scheme, synchronization occurs in real time over some sort of direct communications link. For example, the connection could be over a modem, network, or radio modem. Remote sites connect at intervals of minutes, hours, days, or weeks.

A session-based synchronization process is analogous to a telephone conversation in which all outstanding issues at both ends are resolved. The process follows a particular format. A MobiLink remote site begins by opening a connection to a MobiLink synchronization server and uploading a complete list of all the changes made to the remote database since the previous synchronization. Upon receiving this data, the server updates the consolidated database, then sends back all relevant changes. The remote site incorporates the entire set of changes, then sends back a confirmation and closes the connection.

Message-based replication: SQL Remote

SQL Remote exchanges data between databases using **messages**. Messages are typically files, placed in a particular directory, or specially formatted e-mail messages. A **message agent**, attached to each database, sends messages regarding changes to its own data. The same agent also receives messages from one or more other databases and modifies the database, according to the contents of the received messages. This system allows replication between databases that have no direct connection: an occasional message-based connection such as e-mail or a periodic dial-up link is sufficient.

In message-based communications, each message carries its destination address and other control information so that no direct connection is necessary between applications exchanging information. For example, an e-mail message contains the destination address; there is no direct connection between the sending server and the recipient.

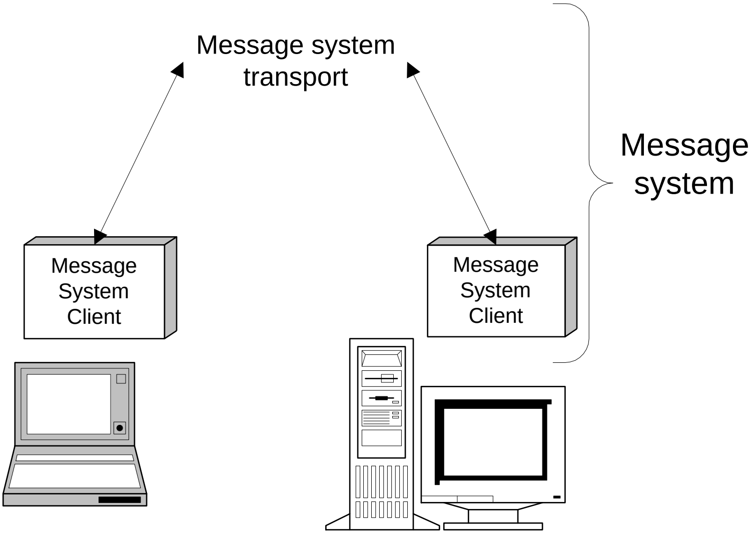
Message services use store and forward methods

Just as session-based client/server applications rely on network communication protocol stacks, such as TCP/IP or Novell NetWare's SPX, so message-based applications rely on message services such as Internet Simple Mail Transfer Protocol (SMTP), Microsoft's Messaging API

(MAPI), Lotus' Vendor Independent Messaging (VIM), or a simple shared file link.

Message services use **store-and-forward** methods to get each message to its destination: for example, e-mail systems store messages until the recipient opens their mail folder to read their mail, at which time the e-mail system forwards the message.

Building a replication system on top of a message system means that a message-based replication system, such as SQL Remote, does not need to implement a store-and-forward system to get messages to their destination. Just as session-based client/server applications do not implement their own protocol stacks to pass information between client and server, so SQL Remote uses existing message systems to pass the messages.



Guaranteed delivery

To work reliably, a message-based replication system must both guarantee that all messages reach their destination and that the messages are applied in the same order that they are sent. SQL Remote incorporates a protocol to guarantee application of replication updates in the correct order.

Connection-based replication: Replication Server

Some replication technologies rely on the presence of a continuous, or at least almost continuous, connection between the databases. Through this connection, the two databases conduct an ongoing dialogue. These types of systems excel at replicating changes quickly. Indeed, given sufficient resources and channel capacity, replication can occur reliably with a lag time of no more than a few seconds.

Replication server is a near real-time replication system designed primarily for replication between a small number of databases. It is normally used with a continuous, reliable, high-speed connection. It incorporates store-and-forward techniques that allow replication to continue automatically when a connection is lost and later re-established.

The main drawback of this type of system is that a reliable, continuous connection can be expensive to maintain. This restriction makes connection-based technologies suited to replication between two large, fixed databases. In environments where the remote machines are mobile or are only occasionally connected, message-based or session-based technologies provide more flexible solutions.

Sybase replication technologies

Sybase provides three replication technologies:

- **MobiLink** is a session-based technology intended for the two-way replication of data between a central, consolidated database and a large number of remote databases. It supports a variety of consolidated database servers, including non-Sybase databases. Administration and resource requirements at the remote sites are minimal, making it well suited to a variety of mobile applications. At the end of each synchronization session, the databases are consistent.
- **SQL Remote** is a message-based technology intended for the two-way replication of transactions. It is designed for two-way replication involving a consolidated data server and large number of remote databases. Administration and resource requirements at the remote sites are minimal, making it well suited to mobile databases. This system is message based. Depending on the setup, typical lag times between the consolidated and remote databases can be on the order of seconds, minutes, or hours.
- **Replication Server** is a connection-based technology intended for the two-way replication of transactions. It is well suited to replication between a small number of enterprise databases connected by a high-speed network, generally with an administrator at each site. In such a setup, it is possible to achieve lag times as low as a few seconds.

Choosing a replication technology

Each Sybase replication technology lends itself to particular applications. The following descriptions differentiate the technologies and let you select the one best suited to your needs.

You should consider which of the following elements are important in your application

Your consolidated database system

In a typical replication environment, a large database serves as a central repository for information. Sometimes you can choose a database system that suits your needs. Other times, a central database already exists and you must adapt the replication system to work with it.

MobiLink can work with many popular database servers, including Adaptive Server Anywhere, Sybase Adaptive Server Enterprise, Oracle, Microsoft SQL Server, and IBM DB2.

In a SQL Remote installation, the central database must be either Adaptive Server Anywhere or Sybase Adaptive Server Enterprise.

Your remote database system

Sybase replication technologies also differ in the types of remote databases that they can support. MobiLink allows your remote database to be either Adaptive Server Anywhere or UltraLite.

SQL Remote supports only Adaptive Server Anywhere remote databases.

Network characteristics

MobiLink and SQL Remote are both well suited to occasionally-connected environments, where remote sites must operate for hours or days in isolation, although more frequent synchronization is possible whenever a network connection is available. In contrast, Replication Server is designed for a continuous connection to allow large amounts of data to be replicated promptly.

MobiLink is session based. A real-time connection is required during synchronization. If this connection is interrupted before synchronization is complete, the process will not complete until the next synchronization. In contrast, SQL Remote relays information via messages, which can be sent or received asynchronously. These messages may take the form of files on a hard disk, or e-mail messages. These messages can be processed whenever they are received, allowing replication to occur incrementally.

Latency

In some situations, it may be important that your information is replicated immediately. In others, replication once or twice a day may suffice. In fact, more frequent replication may be impossible when no network connection is available.

Both MobiLink and SQL Remote are primarily intended for situations where replication occurs infrequently; for example every few hours or days. MobiLink and SQL Remote can both handle more frequent synchronization, but resource and network requirements are greater. However, given sufficient resources, MobiLink synchronizations can occur every few minutes. SQL Remote, when run in continuous mode, allows replication to occur every few seconds.

Replication Server is designed for setups requiring near real-time replication.

The number of remote sites

If you have a very large number of remote users, the best options are MobiLink or SQL Remote. The SQL Remote message-based design allows a typical installation to handle thousands of remote users. MobiLink scalability is limited only by the scalability of the consolidated database-management system. Replication Server is designed for only a few sites.

While these numbers are guidelines, there is no hard limit on the maximum number of remote sites with any of these systems. The actual number depends on the amount of information replicated, the frequency of synchronization, and the design of your application.

Transaction ordering

SQL Remote replicates data by scanning the transaction log and preparing messages, as appropriate, for each transaction. It orders these messages and sends them to the remote or consolidated site. When processing receives messages, SQL Remote always processes them in the same order as they were applied to the other database. When necessary, it automatically delays processing a message until all earlier messages have been applied.

MobiLink, in contrast, works by grouping the results of multiple transactions on the remote database into one set of changes to be applied to the consolidated database. Since synchronization always occurs at a transaction boundary, referential integrity is preserved. The order of the individual changes made during the component transactions is not preserved. However, since uncommitted data is never synchronized, data integrity is preserved.

Achieving data consistency at a particular time

Immediately following each MobiLink synchronization session, the data in the two databases is consistent. The ability to guarantee the consistency of the data at a remote site at a particular point in time is an advantage of MobiLink session-based replication. For example, if it is important that the data at a remote site accurately reflect the data in the consolidated database at a particular time, say 10 o'clock in the morning, this objective can be achieved by synchronizing just prior to this time. As long as the synchronization completes successfully, the currency of the data at the remote site is assured.

When changes to the data are replicated through an exchange of messages, it is difficult to guarantee that the data in a particular remote site is completely consistent with the data in the consolidated site at any particular point in time. For example, sometimes a message is lost in transit. SQL Remote automatically recognizes this fault and resends the message, but such interruptions can cause unexpected delays.

Replication technology characteristics summary

The following table summarizes the characteristics of each replication technology. Following sections expand on the entries in this table.

Replication technology	**Number of databases**	**Connection**	**Latency**	**Volume**	**Database types**
MobiLink	Large	Occasional	Medium	Medium	Heterogeneous
SQL Remote	Large	Occasional	Low	Medium	Homogeneous
Replication Server	Small	Continuous	Low	High	Heterogeneous

MobiLink characteristics

MobiLink is designed for replication installations with the following requirements:

- **Large numbers of databases** MobiLink is designed to support large numbers of remote databases. It can support thousands of remote databases in a single installation.
- **Occasionally connected** MobiLink supports databases that are occasionally connected or indirectly connected to the network on which the server is running. MobiLink scalability is limited only by the scalability of the consolidated database-management system.
- **Medium to high latency** Latency is the lag time between data being entered at one database and being replicated to each database in the installation. Applications typically connect and synchronize at periods of minutes, hours, or days.
- **Low to medium volume** Download information for remote sites is prepared for one remote site at a time. Large amounts of data in a MobiLink system can cause long connection times, since the remote site cannot disconnect until synchronization is complete.
- **Heterogeneous databases** MobiLink supports many of the most popular relational-database systems for use as a consolidated database. The schema of the remote sites can be different from that of the consolidated database because you control the synchronization process by writing scripts.

SQL Remote characteristics

SQL Remote is designed for replication installations with the following requirements:

- **Large numbers of databases** SQL Remote is designed to support a large number of remote databases. It can support thousands of remote databases in a single installation because the messages for many remote sites can be prepared simultaneously.
- **Occasionally connected** SQL Remote supports databases that are occasionally connected or indirectly connected to the network on which the server is running.
- **Low to high latency** High latency means a long lag time between data being entered at one database and being replicated to each database in the

installation. With SQL Remote, replication messages are sent typically at periods of seconds, minutes, hours, or days.

- **Low to moderate volume** As replication messages are delivered occasionally, a high transaction volume at each remote site can lead to a very large volume of messages. SQL Remote is best suited to systems with a relatively low volume of replicated data per remote database. At the consolidated site, SQL Remote can, however, prepare messages efficiently by preparing messages for multiple sites simultaneously.
- **Homogeneous databases** SQL Remote supports Adaptive Server Enterprise and Adaptive Server Anywhere databases. Each database in the system must have a very similar schema.

Replication Server characteristics

Replication Server is designed for replication installations with the following requirements:

- **Small numbers of databases** Replication Server is designed to support replication among servers, with installations typically involving fewer than one hundred servers.
- **Continuously connected** Connections between primary sites and replicate sites may be over a wide area network, but Replication Server is designed for situations where there is a near-continuous connection path for data exchange among the servers in the installation.
- **Low latency** Low latency means a short lag time between data being entered at one database and being replicated to each database in the installation. With Replication Server, replication messages are sent typically within seconds of being entered at a primary site.
- **High volume** With near-continuous connections and high performance, Replication Server is designed for a high volume of replication messages.
- **Heterogeneous databases** Replication Server supports several leading DBMSs, and allows mapping of object names during replication, so that support for heterogeneous databases is provided.

PART II

SQL ANYWHERE STUDIO TUTORIALS

This part contains tutorials to help you get started using all the components of SQL Anywhere Studio.

CHAPTER 4

Tutorial: Connecting to the Sample Database

About this chapter

This chapter provides basic information about starting, running, and stopping database servers. It provides a tutorial that uses a personal server running the sample database.

☞ For complete instructions on connecting to servers and databases, see “Running the Database Server” [*ASA Database Administration Guide*, page 3] and “Connecting to a Database” [*ASA Database Administration Guide*, page 37].

Contents

Introduction

Adaptive Server Anywhere databases are held in files on disk. The Adaptive Server Anywhere database server is the piece of software that manages the database. All access to your database must be made through a database server.

Adaptive Server Anywhere comes with two versions of its database server. The first is called the **personal server**. It can accept connections from applications or users running on the same machine. By contrast, the **network server** also accepts connections from applications or users on other computers by means of the network, which links the two machines. Apart from this difference, these two versions of the database server perform the same tasks and are virtually identical.

The database server allows access to databases from client applications, and processes commands in a secure and efficient manner. Only one database server at a time can manage any one database.

In this tutorial you start the database server running the sample database. You then connect to the database from Interactive SQL and send a command to the database. The tutorial leads you through creating an ODBC data source to hold connection information, and finishes by shutting down the database server.

Timing

The tutorial takes about ten minutes.

Goals

The tutorial provides you with the basic skills of running a database server and connecting to it. These skills are assumed elsewhere in the documentation.

Lesson 1: Start the database server

This section describes how to start the database server running the sample database. In the documentation, *starting a database* is often used as a convenient way of saying *starting a database server running a database* . Depending on the operating system you are using, you have a choice of how to start the database server running the sample database.

❖ Start the database server running the sample database (Windows)

1. From the Start menu, choose Programs ➤ SQL Anywhere 9 ➤ Adaptive Server Anywhere ➤ Personal Server Sample.

   This starts a personal server running the sample database. The server appears as an icon in the system tray, at the opposite end of the Taskbar from the Start button.

❖ Start the database server running the sample database (Command prompt)

1. At a command prompt, change to your SQL Anywhere installation directory. On Windows operating systems, the default installation directory is *C:\Program Files\Sybase\SQL Anywhere 9*.

2. Start the database server running the sample database.

   The way you start the database server depends on your operating system, and on whether you wish to connect to the database from other machines on the network.

   - On Windows or UNIX operating systems, enter the following command:

     ```
     dbeng9 -n asademo9 asademo.db
     ```

   - On Windows or UNIX operating systems, if you wish to connect to the database server from other machines on the network, enter the following command:

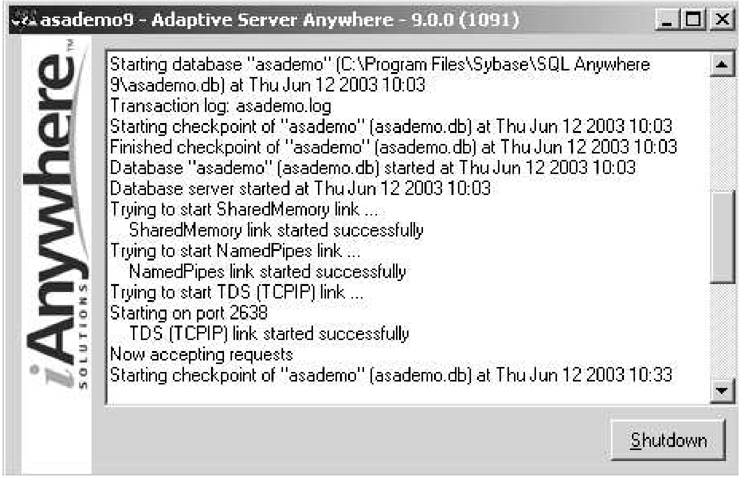
     ```
     dbsrv9 -n asademo9 asademo.db
     ```

   - On NetWare, enter the following command:

     ```
     load dbsrv9.nlm -n asademo9 asademo.db
     ```

   The database server starts.

You can display the database server window in Windows by double-clicking the Adaptive Server Anywhere icon in the system tray at the bottom of your screen.



```
Starting database "asademo" (C:\Program Files\Sybase\SQL Anywhere
9\asademo.db) at Thu Jun 12 2003 10:03
Transaction log: asademo.log
Starting checkpoint of "asademo" (asademo.db) at Thu Jun 12 2003 10:03
Finished checkpoint of "asademo" (asademo.db) at Thu Jun 12 2003 10:03
Database "asademo" (asademo.db) started at Thu Jun 12 2003 10:03
Database server started at Thu Jun 12 2003 10:03
Trying to start SharedMemory link ...
  SharedMemory link started successfully
Trying to start NamedPipes link ...
  NamedPipes link started successfully
Trying to start TDS (TCPIP) link ...
Starting on port 2638
  TDS (TCPIP) link started successfully
Now accepting requests
Starting checkpoint of "asademo" (asademo.db) at Thu Jun 12 2003 10:33
```

The server window displays the following information:

- **The server name** The name in the title bar (in this case **asademo9**) is the **server name**. A server name is assigned each time a database server is started. This name can be used by applications when they connect to a database.

- **Startup information** When a database server starts, it sets aside some memory that it uses when processing database requests. This is called the **cache**. The amount of cache memory appears in the window. The cache is organized in fixed-size **pages**, and the page size also appears in the window.

- **Database information** The names of the database file and its transaction log file appear in the window.

In this case, the startup cache size and page size are the default values. For many purposes, default startup options are fine. For more advanced use, you can provide explicit database server startup options.

☞ For a complete list of startup options, see “The database server” [*ASA Database Administration Guide,* page 124].

Next step

The database server is ready to accept connections from database applications. In the next lesson, you connect to the sample database from the Interactive SQL database administration utility.

Lesson 2: Connect to the sample database from Interactive SQL

Once the database server is running, you can connect to it from an application. Here we connect to it from Interactive SQL, and then test that the connection is working.

❖ Connect to the sample database from Interactive SQL (Windows)

1. Ensure that the sample database is loaded on a running server, as described in “Lesson 1: Start the database server” on page 37.

2. Start Interactive SQL:

   ♦ Choose Start ➤ Programs ➤ SQL Anywhere 9 ➤ Adaptive Server Anywhere ➤ Interactive SQL.

   The Connect dialog appears.

   - Click ODBC Data Source Name. Click Browse, select **ASA 9.0 Sample** from the list, then click OK.
   - Click OK to connect to the sample database.

❖ Connect to the sample database from Interactive SQL (Command prompt)

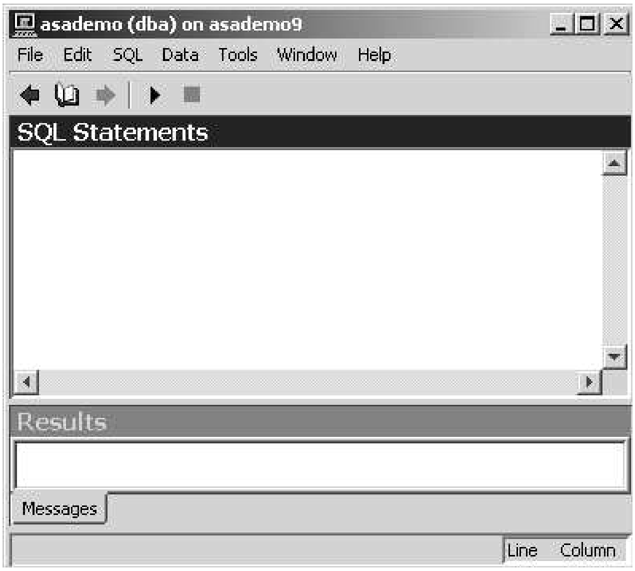
1. Ensure that the sample database is loaded on a running server, as described in "Lesson 1: Start the database server" on page 37.

2. Start Interactive SQL.

   Type the following command:

   ```
   dbisql -c "dsn=ASA 9.0 Sample"
   ```

After connecting to the database, the Interactive SQL window appears.



The database name, user ID, and server name appear in the title bar, confirming that you have successfully connected to the database.

Lesson 3: Send a command to the database

Now you are connected to the database from Interactive SQL, you can send commands to the database. Commands take the form of SQL statements.

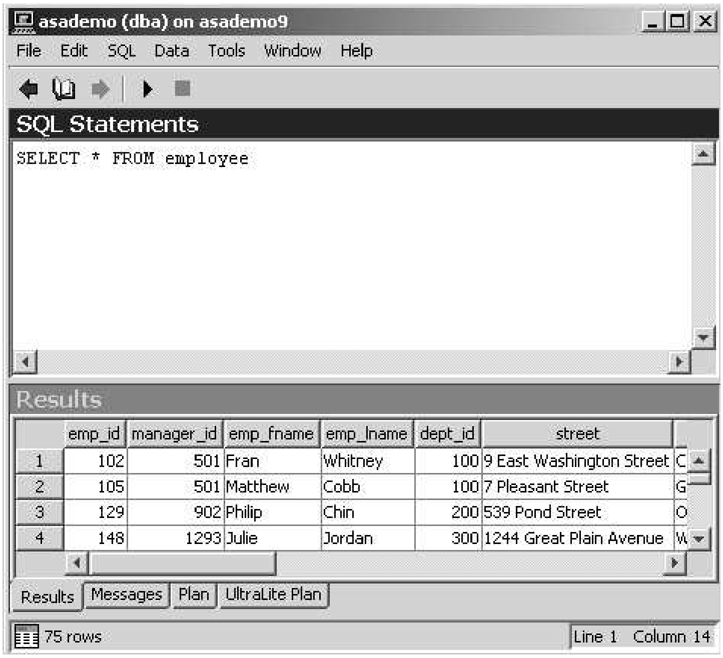
❖ Send a command to the database

1. In the SQL Statements pane, type the following SQL query.

   ```
   SELECT * FROM employee
   ```

2. Press F5 to execute the statement.

   The query displays all the columns of the table named employee. This table contains information about the employees of a fictitious merchandising company. The following appears:



```
SELECT * FROM employee
```

	emp_id	manager_id	emp_fname	emp_lname	dept_id	street	
1	102	501	Fran	Whitney	100	9 East Washington Street	C
2	105	501	Matthew	Cobb	100	7 Pleasant Street	G
3	129	902	Philip	Chin	200	539 Pond Street	O
4	148	1293	Julie	Jordan	300	1244 Great Plain Avenue	W

Interactive SQL retrieves the information by sending a request to your database server. The database server, in turn, looks up the information in the employee table and returns it to Interactive SQL.

At this point, you may want to look at the data in some of the other tables in the sample database, such as product, department, and sales_order.

☞ For more information about Interactive SQL, see "Using Interactive SQL" [*ASA Getting Started*, page 67].

Lesson 4: Create an ODBC data source

The **Open Database Connectivity** (**ODBC**) interface is defined by Microsoft Corporation, and is a standard interface for connecting client applications to database-management systems in Windows environments. Connections are made by specifying connection parameters. It is often convenient to collect a set of connection parameters together and store them in an **ODBC data source**.

You can connect to Adaptive Server Anywhere databases using ODBC data sources from Interactive SQL, Sybase Central, PowerDesigner, InfoMaker, and all the Adaptive Server Anywhere utilities.

Adaptive Server Anywhere supports several programming interfaces in addition to ODBC. You can use ODBC data sources to connect to Adaptive Server Anywhere—not only from ODBC applications, but also from embedded SQL applications, such as the administration utilities, and from Interactive SQL and Sybase Central—even though these applications do not use ODBC. The functionality to process ODBC data sources is built into the embedded SQL client library, and into Sybase Central and Interactive SQL.

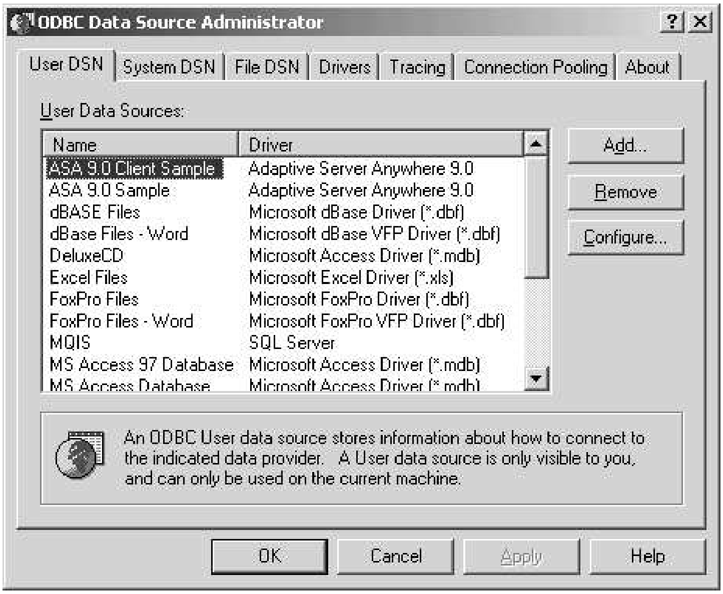
In this lesson, we show how to create an ODBC data source.

❖ Create an ODBC data source

1. Start the ODBC Administrator.

   From the Windows Start menu, choose Programs ➤ SQL Anywhere 9 ➤ Adaptive Server Anywhere ➤ ODBC Administrator.

   The ODBC Data Source Administrator appears, displaying a list of the data sources you currently have installed on your machine:



User Data Sources:

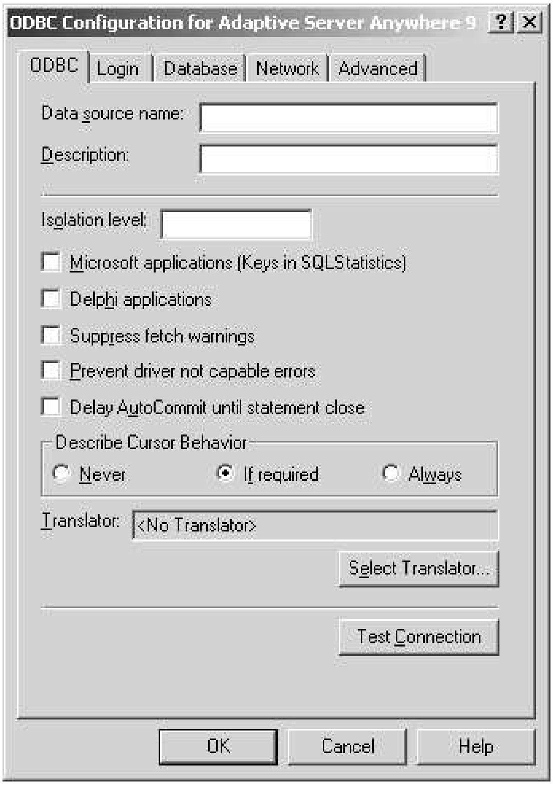
Name	Driver
ASA 9.0 Client Sample	Adaptive Server Anywhere 9.0
ASA 9.0 Sample	Adaptive Server Anywhere 9.0
dBASE Files	Microsoft dBase Driver (*.dbf)
dBase Files - Word	Microsoft dBase VFP Driver (*.dbf)
DeluxeCD	Microsoft Access Driver (*.mdb)
Excel Files	Microsoft Excel Driver (*.xls)
FoxPro Files	Microsoft FoxPro Driver (*.dbf)
FoxPro Files - Word	Microsoft FoxPro VFP Driver (*.dbf)
MQIS	SQL Server
MS Access 97 Database	Microsoft Access Driver (*.mdb)
MS Access Database	Microsoft Access Driver (*.mdb)

An ODBC User data source stores information about how to connect to the indicated data provider. A User data source is only visible to you, and can only be used on the current machine.

2. On the User DSN tab, click Add.

3. Select Adaptive Server Anywhere 9.0 from the list of drivers, and click Finish.

   The Adaptive Server Anywhere ODBC Configuration dialog appears.


ODBC Configuration for Adaptive Server Anywhere 9

ODBC | Login | Database | Network | Advanced

Data source name:

Description:

Isolation level:

☐ Microsoft applications (Keys in SQLStatistics)

☐ Delphi applications

☐ Suppress fetch warnings

☐ Prevent driver not capable errors

☐ Delay AutoCommit until statement close

Describe Cursor Behavior

○ Never ● If required ○ Always

Translator: <No Translator>

Many of the fields in this dialog are optional. Click the question mark at the top right of the window and click an entry field to find more information about that field. For many purposes only need to use a few parameters. The following data source parameters are sufficient for the Adaptive Server Anywhere sample database:

- **Data Source Name (ODBC tab)** This is the name that appears in the Connect dialog. It can contain spaces, but should be short.

  Enter the name **My Sample DSN**.

- **User ID (Login tab)** The database user ID you will use to connect. If you omit the user ID, you will be prompted for it when you attempt to connect.

  The default user ID for Adaptive Server Anywhere databases is **DBA**. Enter this as the user ID.

- **Password (Login tab)** You should omit the password if there are security concerns with having passwords stored on your machine. If you omit the password, you are prompted for it when you attempt to connect.

The default user ID for Adaptive Server Anywhere databases has a password of **SQL**. Enter this as the password.

In production data sources, you may want to leave the user ID and password blank. The user is prompted to enter the information when they connect to the database.

♦ **Database File (Database tab)** You can select a database file by browsing your machine.

Browse to your SQL Anywhere installation directory, and select the *asademo.db* file. This is the Adaptive Server Anywhere sample database.

4. When you have specified the parameters you need, click OK to create the data source and close the dialog box.

5. Start Interactive SQL and connect to the Adaptive Server Anywhere sample database using your new ODBC data source.

☞ For a full description of database connections, see “Connecting to a Database” [*ASA Database Administration Guide,* page 37].

☞ For more information about ODBC data sources, see “Working with ODBC data sources” [*ASA Database Administration Guide,* page 53].

☞ For information about setting up ODBC data sources in UNIX, see “Using ODBC data sources on UNIX” [*ASA Database Administration Guide,* page 63].

Lesson 5: Shut down the database server

You can now shut down the database server you worked with in previous sections.

❖ Shut down the database server running the sample database (Windows)

1. Shut down Interactive SQL.

   In the SQL Statements pane, type `Exit` and press F5 to execute the command.

2. Double-click the Adaptive Server Anywhere icon in the Windows task bar.

3. Click Shutdown.

❖ Shut down the database server running the sample database (UNIX)

1. Shut down Interactive SQL.

   In the SQL Statements pane, type `Exit` and press ENTER.

2. Press **q** in the console where you started the database server.

You should not generally shut down a database server while client applications are still connected. If you attempt to do so, a warning appears. If you ignore the warning and proceed, you lose any pending transactions.

CHAPTER 5

Tutorial: Managing Databases with Sybase Central

About this chapter

This chapter introduces Sybase Central, the Sybase database management tool. It provides a brief introduction to using Sybase Central for managing database properties.

Contents

About Sybase Central

Sybase Central is a database management tool that provides Adaptive Server Anywhere database settings, properties, and utilities in a graphical user interface. Sybase Central can also be used for managing other Sybase products. This chapter describes how to use Sybase Central with Adaptive Server Anywhere databases.

Sybase Central helps administer two types of tasks:

- Tasks carried out by sending SQL statements to the server.
- Tasks carried out by Adaptive Server Anywhere utilities.

Sybase Central provides an easy-to-use interface for both kinds of tasks.

☞ For more information about Adaptive Server Anywhere utilities, see “Database Administration Utilities” [*ASA Database Administration Guide*, page 455].

Lesson 1: Start Sybase Central

This tutorial introduces the Sybase Central user interface. Lesson 1 describes how to start Sybase Central, how to connect to a database, and how to view a database schema in Sybase Central.

The following lessons show you how to create, delete, and view tables, columns, and procedures; how to manage users and user groups; and how to back up your database.

Before you begin, make a copy of the sample database so that you can restore it after you have finished the tutorial.

❖ To copy the sample database

1. Navigate to the directory that contains the sample database file, *asademo.db*. The default installation directory for it is *Program Files\Sybase\Adaptive Server Anywhere 9*.

2. Create a sub-directory to hold a copy of *asademo.db*. For example, call it *demoback*.

3. Create a copy of *asademo.db* in the backup directory. You will not touch this copy, but work with the original version.

Start Sybase Central

You can start Sybase Central either from the Windows Start menu or from a command prompt.

❖ Start Sybase Central (Windows)

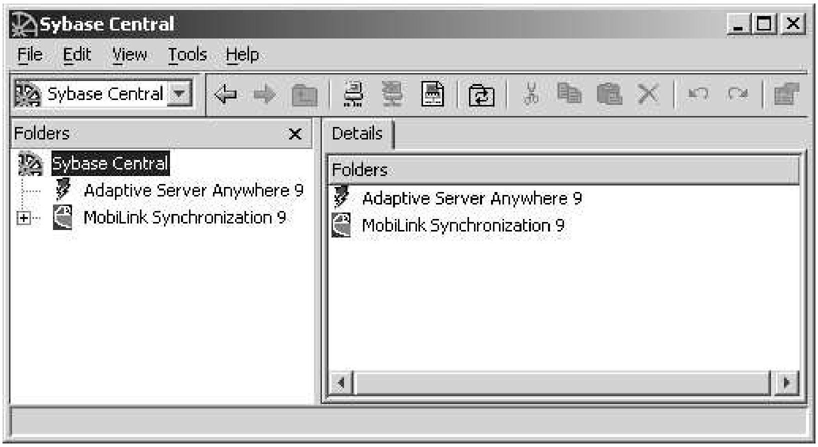
1. Choose Start ➤ Programs ➤ SQL Anywhere 9 ➤ Sybase Central.

❖ Start Sybase Central (Command prompt)

1. At the command prompt, enter the following command:

```
scjview
```

The Sybase Central main window appears:


Sybase Central
File
Edit
View
Tools
Help
Sybase Central
Folders
Sybase Central
Adaptive Server Anywhere 9
MobiLink Synchronization 9
Details
Folders
Adaptive Server Anywhere 9
MobiLink Synchronization 9

The Sybase Central main window is split into two vertically-aligned panes.

The left pane displays a hierarchical view of database objects or **containers** in a tree-like structure. A container is a database object that can hold other database objects, including other containers.

The right pane displays the contents of the container that is selected in the left pane. You can change the appearance of both panes in the Options dialog (accessed through the Tools menu).

Toolbar

The main dialog's toolbar provides you with buttons for common commands. To display or hide the toolbar, choose View ➤ Toolbar.

With the main toolbar, you can:

- Navigate through the object tree.
- Connect to or disconnect from a database, server, or product module.
- Access the Connection Profiles dialog (also accessible through the Tools menu).
- Refresh the view of the current folder. You can also press F5 to refresh the folder.
- Cut, copy, paste, or delete objects.

♦ View the property sheet of a selected object.

As an easy reminder of what these buttons do, you can hold your cursor over each button to see a popup description.

Status bar

The status bar, which appears at the bottom of the main window, displays a brief summary of menu commands as you navigate through the menus. To display or hide the status bar, choose View ➤ Status Bar.

Connect to the sample database from Sybase Central

This section describes how to connect to the sample database from Sybase Central.

By default, all newly-created Adaptive Server Anywhere databases contain a user ID of **DBA** and a password of **SQL**. It is the responsibility of the database administrator to provide the desired level of security by changing passwords and creating other user IDs.

The user ID and password for the sample database are also DBA and SQL. However, when you connect using the ASA 9.0 Sample data source (as the procedure below describes), you do not need to enter this information because the data source already contains it.

❖ To connect to the sample database

1. Start Sybase Central.
2. Choose Tools ➤ Connect.
3. If a dialog prompts you to choose a plug-in, choose Adaptive Server Anywhere 9 and click OK.

   The Connect dialog opens.
4. Select the ODBC Data Source Name option.
5. In the ODBC Data Source Name box, type ASA 9.0 Sample, or select it from the dropdown list if it is already there, and click OK. Alternatively, you could also click Browse and choose the ASA 9.0 Sample from the Data Source Names dialog.
6. Click OK to connect to the sample database.

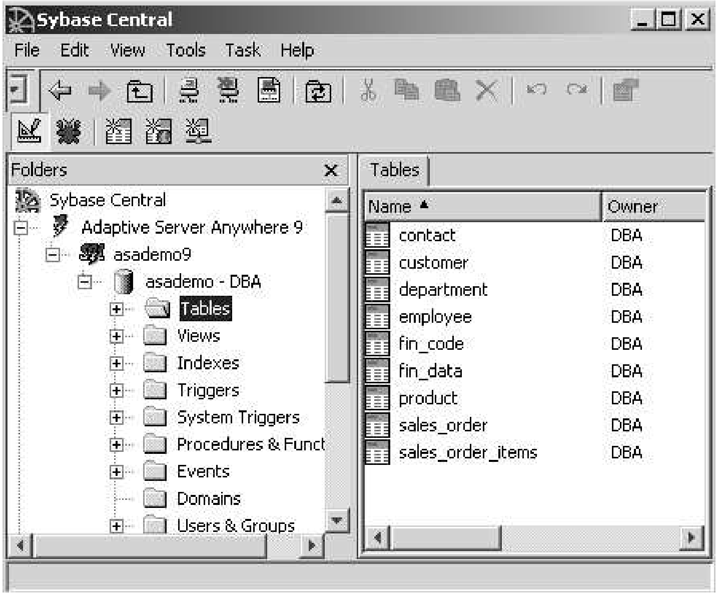
Explore the database schema

The database **schema** is a collection of all the objects in a database. Sybase Central displays a database schema as a hierarchy of containers and their contents. This section describes how to view the schema of a database.

Expanding a database container

Just as in the Windows Explorer, Sybase Central offers a variety of methods for viewing the objects in a database, including:

- Click a container in the left pane to select that container. The right pane then displays the contents of the selected container.
- Click once on the plus or minus icon next to a container in the left pane. This action expands or collapses the container. If no plus or minus icons appear next to a container, it means that the container holds no objects extending beyond the level of that container.
- Double-click a container in either pane. This action expands the tree in the left pane and displays the contents of the container in the right pane.



Name ▲	Owner
contact	DBA
customer	DBA
department	DBA
employee	DBA
fin_code	DBA
fin_data	DBA
product	DBA
sales_order	DBA
sales_order_items	DBA

The left pane displays container objects only. The right pane displays the contents of the container object selected in the left pane. For example, when you select the Tables folder in the left pane, all of the tables within that folder appear in the right pane.

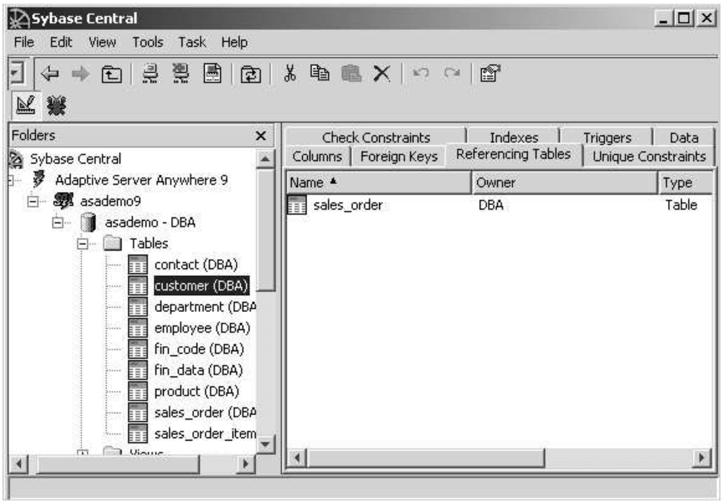
Viewing the tables in a database

The following illustrates the steps taken to examine the contents of a table folder in a database container.

❖ To examine the tables in the sample database

1. Expand the Tables folder. You may have to expand asademo first.

2. Each table in the Tables folder is itself a container. Select a table in the left pane.



Name ▲	Owner	Type
sales_order	DBA	Table

3. Click on the various tabs in the right pane to see the contents of that tab as it relates to the selected table.

Viewing database object properties

The properties of database objects, such as a database or a stored procedure, can be viewed using any of the following methods.

- Right-click a database object in the left pane and choose Properties from the popup menu.

- Select a database object and choose File ➤ Properties.

You can navigate through a database schema by clicking or double-clicking items.

Database folders

Every Adaptive Server Anywhere database contains the following folders:

- **Tables** base tables and global temporary tables stored in the database.

- **Views** computed tables, stored in the database as a query and evaluated when accessed.

- **Indexes** indexes in the database that provides an ordering of the rows of a table on the basis of the values in some or all of the columns

- **Triggers** special form of stored procedure executed automatically when a query that modifies data is executed

- **System Triggers** triggers that implement referential integrity actions in the database

- **Procedures & Functions** for using a module-based language consisting of SQL procedures.

- **Events** for creating and editing events.

- **Domains** for creating non-standard data types.

- **Users & Groups** for administering who is permitted to use the database.

- **Integrated Logins** for enabling users to connect to a database using their Windows NT user name and password.

- **SQL Remote Users** for administering SQL Remote replication of data in the database.

- **MobiLink Users** for managing MobiLink users and data synchronization.

  Publications database objects describing data to be replicated.

- **UltraLite Projects** for collecting SQL statements and table definitions used in UltraLite applications.

- **Dbspaces** for creating more than one *.db* file for the database.

- **Remote Servers** for identifying remote servers so that local users can execute remote procedure calls or retrieve information from the remote server.

  Web Services a list of web services currently available in the database

You should explore the sample database until you are comfortable locating database objects in the Sybase Central main window.

View the data in the database

In addition to viewing the database schema, you can view the data in the database tables from Sybase Central.

❖ To view the contents of database tables

1. Expand the Tables folder.

2. In the left pane, click the contact table.

3. In the right pane, click the Data tab. The data in the table appears in the right pane.

Lesson 2: Create and edit tables

This lesson takes you through the steps required to add a table to the sample database. This task includes adding columns to an existing table.

Create a table in Sybase Central

In this lesson, we create a table of offices in the sample database.

❖ Create a table named office

1. In the left pane of Sybase Central, click the Tables folder.
2. Click the New Table button in the toolbar.

   You can hold your mouse over the buttons to see popup descriptions of the toolbar buttons.

   The Table Creation wizard appears.
3. Follow the instructions in the wizard, naming your table **office**, and choosing not to assign the table a primary key. Use the default settings for the rest of the pages.
4. Name the column that appears in the right pane of Sybase Central when the wizard closes.

Edit the Office table

This section describes how to edit an existing table. The steps show you how to add a primary key to the table created in the previous section by turning a column named office_id into a primary key.

❖ Edit the Office table

1. Open the Tables folder and select the office table.
2. Give the first row the name of office_id, and assign a data type of small int.

   The checkmark in the PKey column indicates that the column named office_id is the primary key for the table.

   > **Primary key conditions**
   > If a checkmark appears in the Nulls column, or if duplicate values are stored in the office_id column, the column cannot be changed to a primary key.
3. On the toolbar, click Save Table.

4. On the toolbar, click Add Column.

5. Edit the new column so that its name is office_name, its type is char, and its size is 20. Select Nulls so that office_name is allowed to be blank.

6. Click Save Table.

The column is now in the database, although it contains no data.

Notes

- Most objects in Sybase Central have property sheets, including tables, users, and stored procedures.

- When you click on many of the folders, **New *item*** buttons appear in the toolbar. Clicking these buttons creates new objects for the database.

- Whenever you select an object in Sybase Central, commands related to that object appear in several places: in the File menu, in a popup menu that you can access by right-clicking the object, and as buttons on the toolbar.

- Once you open a container, its contents are cached to improve performance. You can access two refresh commands in the View menu to update the view of either the current container or the entire window.

Delete the office table

Tables can be deleted, or dropped, from a database. For example, after creating and editing the office table in the previous tutorials, you can delete this table to restore the sample database to its original state.

❖ To delete the office table from the sample database

1. Open the Tables folder.

2. Right-click the office table and choose Delete from the popup menu.

   The Confirm Delete dialog appears.

3. Click Yes to delete the office table from the database.

Lesson 3: Manage users and groups

In Adaptive Server Anywhere, both users and groups are objects within the database. However, groups are also containers, capable of containing users and other groups. When users are contained within a group, they are members of that group.

In this structure, permissions granted to a group are inherited by all users and groups contained within. Adaptive Server Anywhere allows you to create users and groups with permission to use a database and grant membership to other groups. Users and groups can be members of multiple groups.

This tutorial demonstrates how to create a group for the database, create an individual user, and make the user a member of the group.

Add a group to the sample database

This section describes how to add a group to the sample database.

❖ To add a group to the sample database

1. Connect to the sample database.
2. Select the Users & Groups folder in the left pane.
3. Click New Group in the toolbar.

   The Group Creation wizard appears.
4. Follow the instructions in the wizard, naming the new group **Sales**, and leaving the rest of the options at their default settings.

The Sales group now appears in both panes.

Add a user to the database

This section describes how to add a user to the sample database.

❖ To add a user to the sample database

1. Connect to the sample database.
2. Select the Users & Groups folder in the left pane.
3. Click New User in the toolbar.
4. Follow the instructions in the User Creation wizard, naming the new user **Sandy**, and using the name **Sandy** as the password. Leave the rest of the settings at their default values.

An icon for the new user appears in both panes.

Add a user to a group

This section describes how to add two users to a group by copying and pasting them.

❖ To add users to a group

1. Open the Users & Groups folder in the left pane.
2. Click the Sales group.
3. Click the Members tab in the right pane.
4. Choose File ➤ New ➤ Members.
5. Choose the user Sandy, then click OK.

Lesson 4: View and edit stored procedures

The Procedures & Functions folder holds stored procedures for a database. This lesson shows how to view and alter the contents of a procedure, as well as how to modify the procedure's properties.

Viewing Stored Procedures

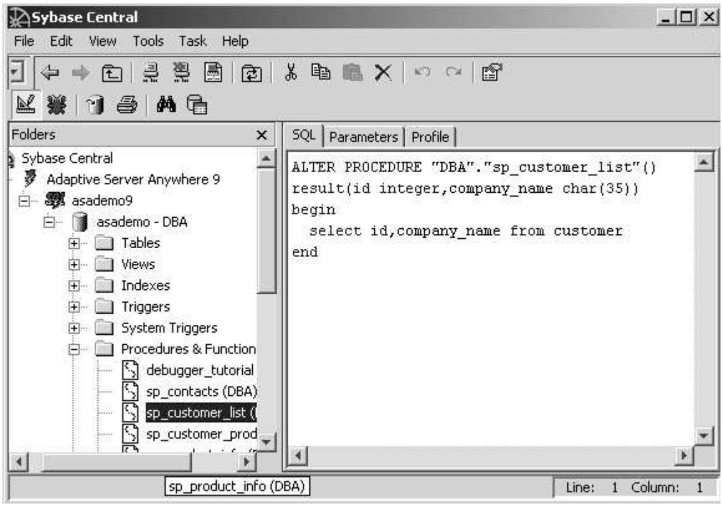
There are several stored procedures included in the sample database. The following steps explain how to view and edit stored procedures.

❖ To view the contents of a stored procedure

1. In the left pane of Sybase Central, open the Procedures & Functions folder in the sample database.

2. Click the sp_customer_list stored procedure.

   You can view the text of the procedure called sp_customer_list on the SQL tab in the right pane:



```
ALTER PROCEDURE "DBA"."sp_customer_list"()
result(id integer,company_name char(35))
begin
  select id,company_name from customer
end
```

   This procedure is designed to return a set of customer IDs and company names from the customer table.

Edit stored procedure properties

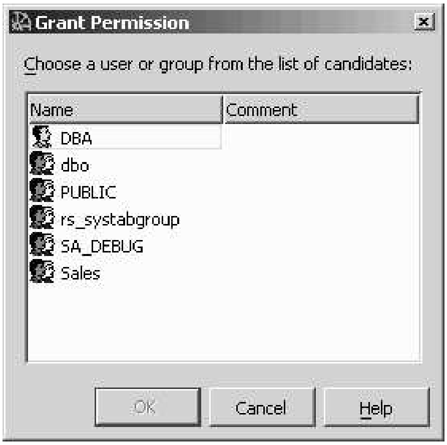
You can edit properties of stored procedures using the stored procedure's property sheet. In this section, we alter the permissions for a stored procedure so that all salespeople, including the user you created in the previous lesson (Sandy), can execute the procedure to obtain a list of customers.

Stored procedures have permissions associated with them. In order to execute a procedure, you either need to be granted permission to execute it, or you need to be a member of a user group that has permission to execute it.

❖ View and alter the permissions on the sp_customer_list procedure

1. In the Procedures & Functions folder, select sp_customer_list.
2. Choose File ➤ Properties.
3. Click the Permissions tab of the property sheet to see which user IDs have been granted permissions for this procedure. Currently, none have since the only user for the sample database is DBA, who is the owner of the procedure and so automatically has execute permission for the procedure.
4. Click Grant.

   The Grant Permissions dialog appears:



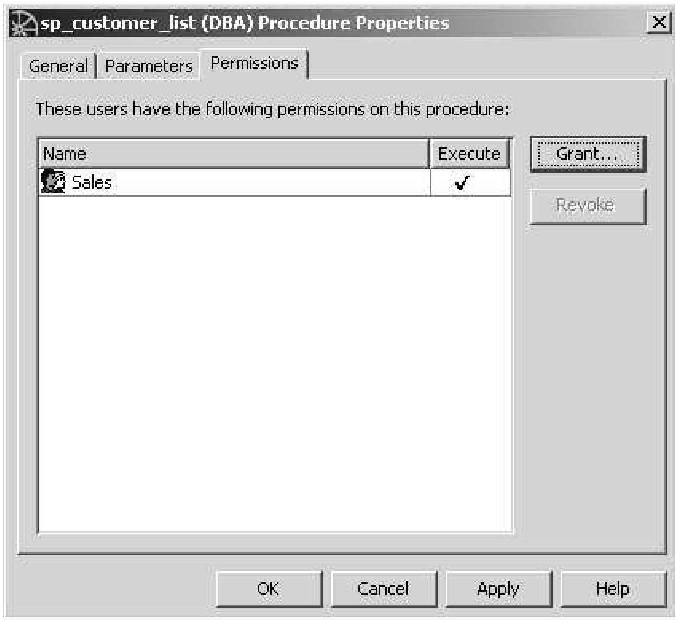
Grant Permission

Choose a user or group from the list of candidates:

Name	Comment
DBA	
dbo	
PUBLIC	
rs_systabgroup	
SA_DEBUG	
Sales	

5. In the Grant Permission dialog, select Sales and click OK to grant this group permission to execute the sp_customer_list procedure.

   Sales is added to the Permissions tab of the Procedure property sheet with a checkmark in the Execute column:



These users have the following permissions on this procedure:

Name	Execute
Sales	✓

You can revoke this group’s permission by clicking the checkmark to make it disappear, or by selecting the group and clicking Revoke at the side of the Permissions tab.

6. Click OK to accept the changes to the sp_customer_list permissions.

Lesson 5: Back up your database

Sybase Central includes a set of database utilities for carrying out common database administration tasks. Wizards walk you through the steps involved.

Utilities

To see a list of those utilities that can be used on a running database, right-click on the sample database container. Several options appear on the popup menu, including Backup Database, Upgrade Database, and Validate Database. Each of these menu items represents a utility.

In this section, we use a wizard to back up the sample database. This can be done on a running database.

❖ To back up a running database

1. In the left pane, right-click the asademo - DBA icon and choose Backup Database from the popup menu. The Backup Database wizard appears.

2. Read the introductory page of the wizard and click Next.

3. Select asademo from the list of databases you can back up. Click Next.

4. Select On Disk, In The Following File.

   Type a filename in the text box indicating where you want to back up the database to. As this is a tutorial, you may wish to choose a file in a temporary directory such as *c:\temp\backup*.

5. Click Finish to back up the database.

Notes

This kind of backup is called an **archive backup**. You can also make backups called image backups, which are physical copies of the database file and associated files. For more information, see “Types of backup” [*ASA Database Administration Guide*, page 346].

An extension of *.1* is added to the filename you specify in the Backup Database wizard.

Wizards are available for several other database administration tasks. You may wish to try creating a database by right-clicking the server in the left pane and choosing Create Database from the popup menu.

Restore the sample database

Now that you have completed this tutorial, you should restore the sample database so that it can be used again. In "Lesson 1: Start Sybase Central" on page 49, you created a copy of *asademo.db*. You can now replace the version of *asademo.db* that you just changed with the copy. Delete *asademo.log*.

If you did not create a copy of *asademo.db*, you can restore the sample database by deleting the Sales group and the user Sandy.

❖ To delete users and groups

1. Select the Users & Groups folder in the left pane.

2. Right-click the icon of the user or group in the right pane.

3. Choose Delete from the popup menu and acknowledge the deletion.

Summary

In this tutorial, you learned how to

- Start Sybase Central and connect to a database.
- Create, modify, and view tables and columns.
- View procedures.
- Manage users and user groups.
- Back up your database.

Where do I go from here?

☞ For more information, see “Working with Database Objects” [*ASA SQL User’s Guide,* page 25].

CHAPTER 6

Tutorial: Designing Databases with PowerDesigner

About this chapter

SQL Anywhere Studio includes PowerDesigner Physical Architect, a module of Sybase's powerful database design tool, PowerDesigner. This module provides ways to generate and modify databases using a graphical representation of the database structure. You can optimize your database by customizing tables, columns, indexes, referential integrity, views, physical storage, triggers, and stored procedures.

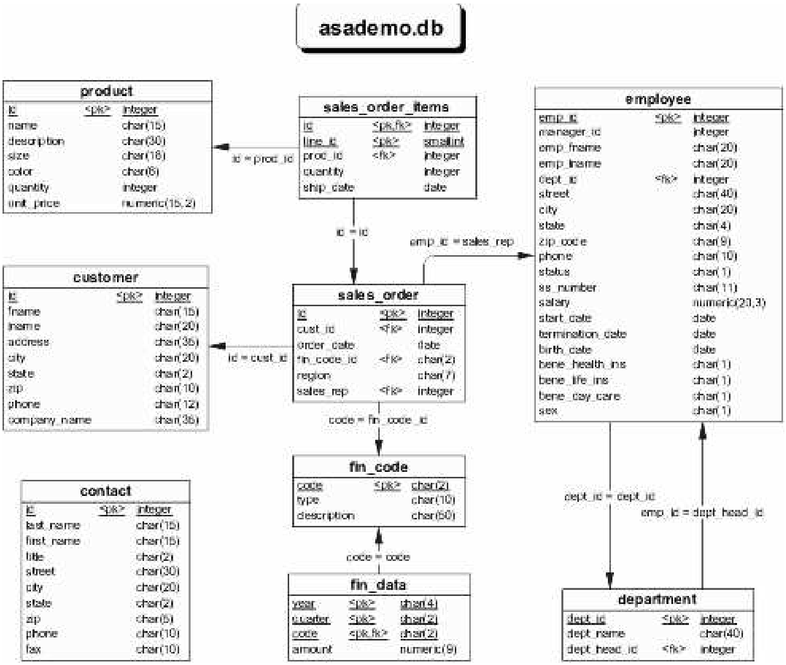
Contents

About PowerDesigner

The structure of your database, such as the tables, relationships, views and triggers, is called the database **schema**. You use SQL statements to create and arrange these elements to your liking, but doing so without a graphical tool can be confusing.

PowerDesigner gives you a graphical representation of the structure of your database. Better, you can modify the structure of the database or create an entirely new one simply by drawing new tables or entering information. Once your design is complete, PowerDesigner can generate a SQL script to generate your new database.

The following diagram, which displays the structure of the sample database, is easily created using PowerDesigner.



product

id	<pk>	integer
name		char(15)
description		char(30)
size		char(18)
color		char(6)
quantity		integer
unit_price		numeric(15,2)

sales_order_items

id	<pk,fk>	integer
line_id	<pk>	smallint
prod_id	<fk>	integer
quantity		integer
ship_date		date

employee

emp_id	<pk>	integer
manager_id		integer
emp_fname		char(20)
emp_lname		char(20)
dept_id	<fk>	integer
street		char(40)
city		char(20)
state		char(4)
zip_code		char(9)
phone		char(10)
status		char(1)
ss_number		char(11)
salary		numeric(20,3)
start_date		date
termination_date		date
birth_date		date
bene_health_ins		char(1)
bene_life_ins		char(1)
bene_day_care		char(1)
sex		char(1)

customer

id	<pk>	integer
fname		char(15)
lname		char(20)
address		char(35)
city		char(20)
state		char(2)
zip		char(10)
phone		char(12)
company_name		char(35)

sales_order

id	<pk>	integer
cust_id	<fk>	integer
order_date		date
fin_code_id	<fk>	char(2)
region		char(7)
sales_rep	<fk>	integer

fin_code

code	<pk>	char(2)
type		char(10)
description		char(50)

contact

id	<pk>	integer
last_name		char(15)
first_name		char(15)
title		char(2)
street		char(30)
city		char(20)
state		char(2)
zip		char(5)
phone		char(10)
fax		char(10)

fin_data

year	<pk>	char(4)
quarter	<pk>	char(2)
code	<pk,fk>	char(2)
amount		numeric(9)

department

dept_id	<pk>	integer
dept_name		char(40)
dept_head_id	<fk>	integer

The performance of your database depends heavily on your design. In general, you should store information about different distinct types of objects, such as employees or products, in separate tables.

You can identify relationships between these tables using references, meaning that foreign keys in one table identify particular rows in another table. Many-to-one and one-to-many relationships can be represented by a reference. Many-to-many relationships require two references and another table.

☞ For more information about database design, see “Designing Your Database” [*ASA SQL User’s Guide*, page 3].

☞ For more information about PowerDesigner, click Help in the toolbar of the PowerDesigner main window to access three books: *PowerDesigner General Features Guide*, *PowerDesigner PDM User’s Guide*, and *PowerDesigner Report User’s Guide*.

☞ For more PowerDesigner tutorials, click Start ➤ Programs ➤ SQL Anywhere 9 ➤ PowerDesigner 8 ➤ PDM Getting Started. This accesses the book *Physical Data Model Getting Started*.

Lesson 1: Getting Started

PowerDesigner can read the structure of a database from a script file that creates the database. However, it is generally easier to just connect to your database from PowerDesigner and let it extract the design directly with the reverse engineering feature.

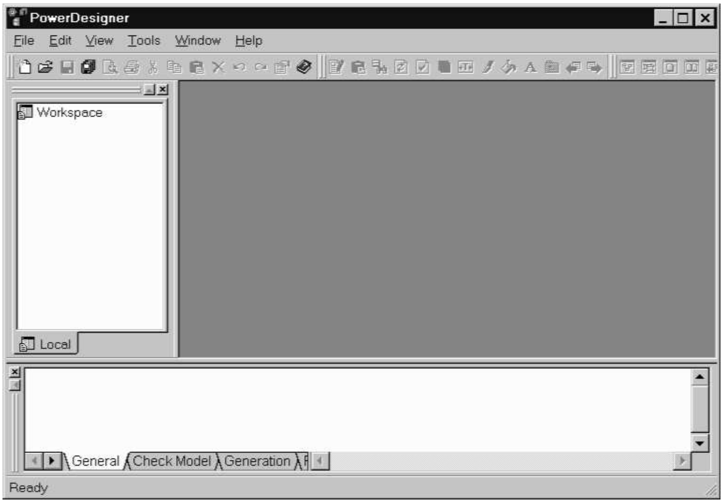
The following tutorial uses the sample database as a starting point. The tutorial illustrates PowerDesigner by implementing a modification that improves the design of the sample database.

Currently, the price of each product is always read from the product table. As a result, updating the price effectively changes the sale price of that item on all previous orders. Adding a unit_price column to the sales_order_items table will correct this problem. The actual selling price to each customer can now be stored separately. The price in the product table records the current list price.

❖ To start PowerDesigner

1. From the Start menu, choose Programs ➤ SQL Anywhere 9 ➤ PowerDesigner 8 ➤ PowerDesigner.

   The PowerDesigner main window appears:

   The PowerDesigner main window includes an object browser docked to the left, and an output window docked at the bottom.

2. Choose File ➤ New.

3. In the New Dialog, choose Physical Data Model and click OK.

The Choose DBMS dialog appears.

4. From the dropdown list, choose Sybase AS Anywhere 9.

5. Ensure Link is selected and click OK.

   The model name MODL_1 appears in the browser and in the title bar.

Reverse engineer the database

In this section, you generate a Physical Data Model (PDM) of the sample database by reverse engineering it.

❖ To reverse engineer the database

1. Click the diagram window (the large central pane of PowerDesigner).

2. From the Database menu in the toolbar, choose Reverse Engineering Database.

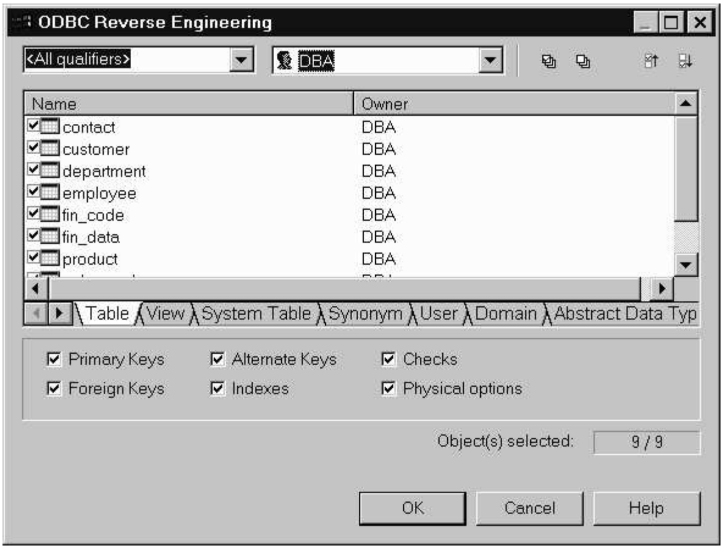
   The Reverse Engineering a Database dialog appears:

3. Ensure that Using an ODBC Data Source is selected, and that the data source is ASA 9.0 Sample.

   If ASA 9.0 Sample does not appear, click the icon to the right of the data source. The Connect to an ODBC Data Source dialog appears. Select Machine Data Source and select ASA 9.0 Sample from the dropdown list. Enter user ID **DBA** and password **SQL**. Click Connect to return to the Reverse Engineering a Database dialog.

4. In the Reverse Engineering a Database dialog, click OK.

   The ODBC Reverse Engineering dialog appears:



Name	Owner
☑ contact	DBA
☑ customer	DBA
☑ department	DBA
☑ employee	DBA
☑ fin_code	DBA
☑ fin_data	DBA
☑ product	DBA

☑ Primary Keys ☑ Alternate Keys ☑ Checks

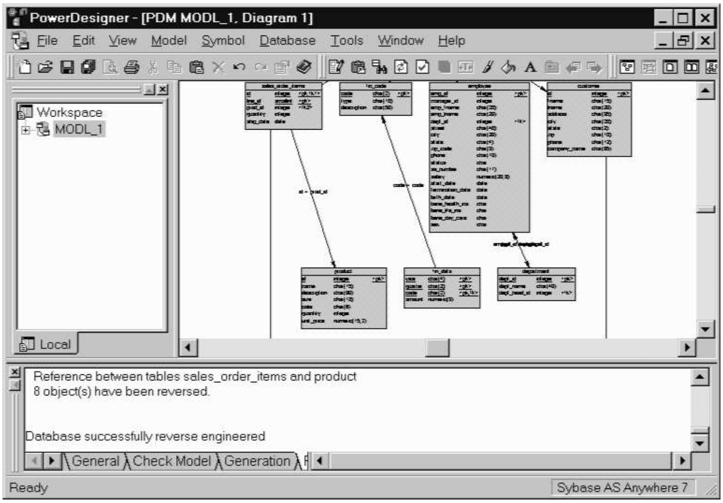
☑ Foreign Keys ☑ Indexes ☑ Physical options

Object(s) selected: 9 / 9

In the lower half of the dialog, there are six checkboxes for selecting keys, indexes, and so on. These are the **reverse engineering options**. Ensure all are selected (the default). You should also ensure that all tables are selected (also the default).

5. Click OK to reverse engineer the database.

   A graphic representation of the sample database appears in the diagram window, and the model objects appear in the browser:



Reference between tables sales_order_items and product
8 object(s) have been reversed.

Database successfully reverse engineered

6. Examine the diagram. You can use F6 to zoom in and F7 to zoom out to make the tables more readable.

- Experiment with these toolbar buttons:

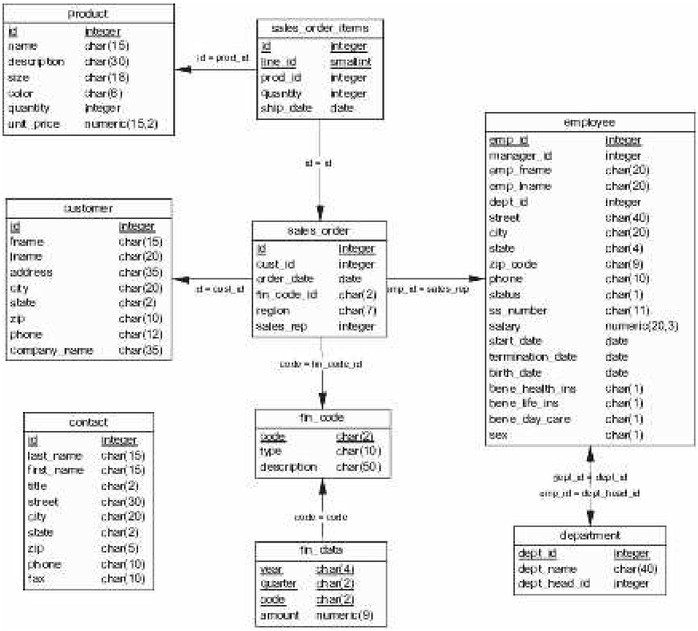
- Choose Tools ➤ Display Preferences to see ways to modify the display. Click each option on the left to see pages of display preference parameters. Click the Help button at the bottom of each page for complete information.

Each table in the database is represented by a box. The name of the table appears at the top of the box. Below is a list of the columns. Column names that are underlined are part of the primary key for that table. The data type of each column appears on the right. Some of the tables may overlap after reverse engineering, but you can use the Auto-layout tool in the Symbol menu to reorganize them.

References between tables are represented with arrows. The arrows point toward the parent table, which is the table that contains the primary key. An equation appears next to each arrow that identifies the reference.

7. Rearrange the tables so that the diagram is easier to read. To do this, drag them with your mouse. The reference arrows follow automatically. You can select multiple objects by holding down the SHIFT key.

   One possible arrangement appears below:



product	
id	integer
name	char(15)
description	char(30)
size	char(18)
color	char(6)
quantity	integer
unit_price	numeric(15,2)

sales_order_items	
id	integer
line_id	smallint
prod_id	integer
quantity	integer
ship_date	date

customer	
id	integer
fname	char(15)
lname	char(20)
address	char(35)
city	char(20)
state	char(2)
zip	char(10)
phone	char(12)
company_name	char(35)

sales_order	
id	integer
cust_id	integer
order_date	date
fin_code_id	char(2)
region	char(7)
sales_rep	integer

employee	
emp_id	integer
manager_id	integer
emp_fname	char(20)
emp_lname	char(20)
dept_id	integer
street	char(40)
city	char(20)
state	char(4)
zip_code	char(9)
phone	char(10)
status	char(1)
ss_number	char(11)
salary	numeric(20,3)
start_date	date
termination_date	date
birth_date	date
bene_health_ins	char(1)
bene_life_ins	char(1)
bene_day_care	char(1)
sex	char(1)

contact	
id	integer
last_name	char(15)
first_name	char(15)
title	char(2)
street	char(30)
city	char(20)
state	char(2)
zip	char(5)
phone	char(10)
fax	char(10)

fin_code	
code	char(2)
type	char(10)
description	char(50)

fin_data	
year	char(4)
quarter	char(2)
code	char(2)
amount	numeric(9)

department	
dept_id	integer
dept_name	char(40)
dept_head_id	integer

Lesson 2: Add a column

You are now ready to add the unit_price column to the sales_order_items table. You can accomplish this task by accessing the list of columns through the Table property sheet.

❖ To add a column

1. Display the column properties:

   ♦ Right-click the sales_order_items table and select Properties from the popup menu.

   The Table property sheet appears:

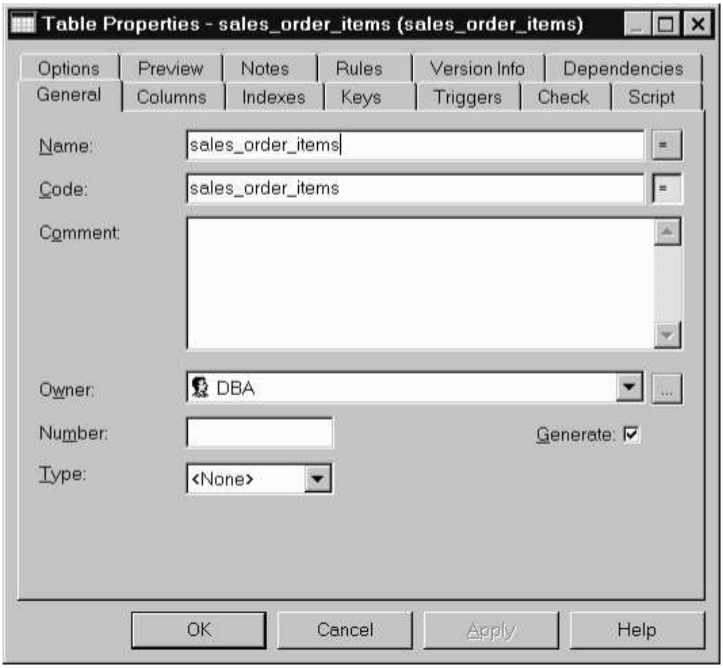


Table Properties - sales_order_items (sales_order_items)

Name: sales_order_items

Code: sales_order_items

Comment:

Owner: DBA

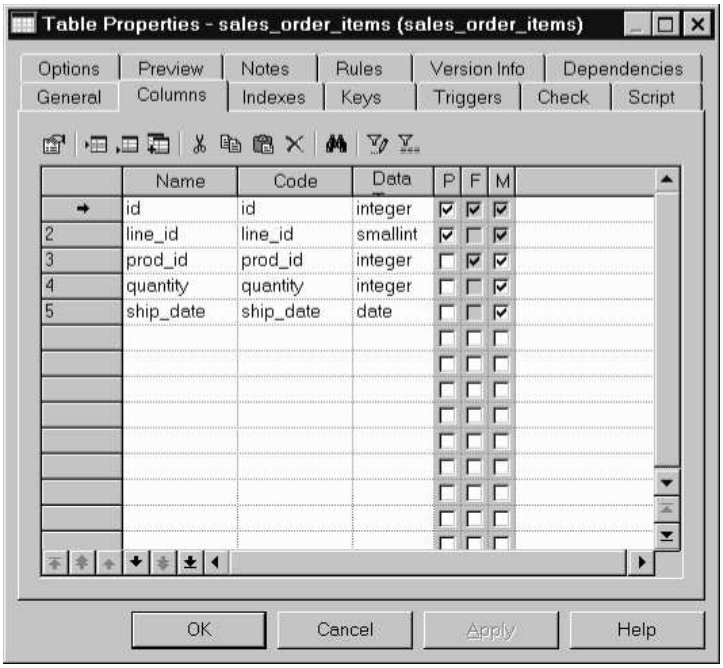
Number:

Generate: ☑

Type: <None>

   ♦ Click the Columns tab.

   The list of columns appears:



	Name	Code	Data	P	F	M
→	id	id	integer	☑	☑	☑
2	line_id	line_id	smallint	☑	☐	☑
3	prod_id	prod_id	integer	☐	☑	☑
4	quantity	quantity	integer	☐	☐	☑
5	ship_date	ship_date	date	☐	☐	☑

2. Add a new column to hold the unit price.

   ♦ Click the Insert a Row tool.

   An arrow appears at the beginning of the line and a column with a default name (Column_6) appears.

   ♦ Type **unit_price** in the name column. The name is automatically duplicated as the code.

   ♦ From the Data dropdown list, choose Numeric. (The Data field may be too narrow to read. You can pull the sides of the column to expand it.)

   ♦ The column properties P, F, and M stand for Primary Key, Foreign Key, and Mandatory, as follows:

   - Primary key designates a column whose values uniquely identify a row in the table.
   - Foreign key designates a column that depends on and migrates from a primary key column in another table.
   - Mandatory indicates a column that must be assigned a value.

   ♦ Select Mandatory and click OK.

3. Examine the effect of your changes on the diagram of the database. The sales_order_items table now includes a new column called unit_price.

Lesson 3: Check your work

PowerDesigner lets you quickly detect database design errors in your new model.

❖ To check your new schema

1. Choose Tools ➤ Check Model.

   The Check Model Parameters dialog appears. You can use the default parameters.

2. Click OK.

   The results of Check Model appear in the Results List.

Lesson 4: Save changes and generate database

In PowerDesigner, models that depict the physical components of your database design, including tables and columns, are called **Physical Data Models** (**PDM**). PowerDesigner stores these in files with the extension *.PDM*.

❖ To save the physical data model (PDM)

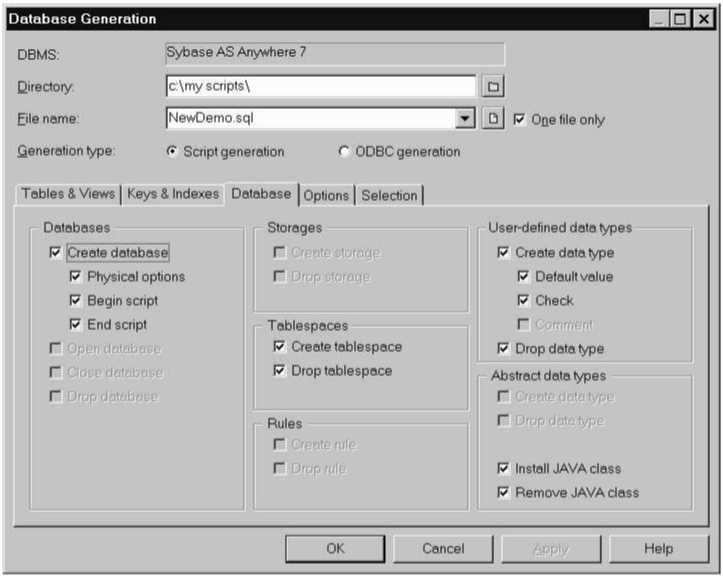
1. Choose File ➤ Save As.
2. Type a file name with the extension *.PDM* and select a location.
3. Click OK.

You can use PowerDesigner to generate a SQL script that implements all the components of your model. You can then use the SQL script to generate a database.

❖ To generate a SQL script to create your new database

1. Choose Database ➤ Generate Database.

   The Database Generation dialog appears:



Database Generation

DBMS: Sybase AS Anywhere 7

Directory: c:\my scripts\

File name: NewDemo.sql ☑ One file only

Generation type: ● Script generation ○ ODBC generation

Databases

☑ Create database

☑ Physical options

☑ Begin script

☑ End script

□ Open database

□ Close database

□ Drop database

Storages

□ Create storage

□ Drop storage

Tablespaces

☑ Create tablespace

☑ Drop tablespace

Rules

□ Create rule

□ Drop rule

User-defined data types

☑ Create data type

☑ Default value

☑ Check

□ Comment

☑ Drop data type

Abstract data types

□ Create data type

□ Drop data type

☑ Install JAVA class

☑ Remove JAVA class

2. Type **NewDemo.SQL** in the File Name box and choose a convenient directory.
3. Ensure that Script Generation is selected.

4. Click the Database tab, and ensure that Create Database is selected.

   Explore the other tabs to observe options that give you control over many other properties of the generated script.

5. Click OK.

   When the script is created, the Result dialog appears.

6. Click Edit to view the script.

   Check that your changes are reflected in the script. For example, the definition of the new office table appears below.

```
/* ========================================= */
/*   Table: office                           */
/* ========================================= */
create table office
(
   id            integer         not null
      default autoincrement
      check (
         id >= 100),
   name          char(15)        not null,
   street        char(30)        not null,
   city          char(20)        not null,
   state         char(2)         not null,
   zip           char(5)         not null,
   phone         char(10)                ,
   fax           char(10)                ,
   primary key (id)
);
```

7. When you are finished, close the dialog: in the Result dialog, click Close.

You can now create your new database from Interactive SQL.

❖ To create the new database

1. Start Interactive SQL.

2. Connect to the sample database. You can use the same ODBC connection.

3. Create an empty database:

   ♦ Execute the following SQL statement, substituting any convenient directory.

```
CREATE DATABASE 'c:\\My Scripts\\newdemo.db'
```

> **Tip**
> To execute a SQL statement in Interactive SQL, type or copy the statement in the SQL Statements pane, and then press F5, *or* choose SQL ➤ Execute.

4. Close the connection to the sample database by clicking SQL ➤ Disconnect.

5. Connect to the new database by clicking SQL ➤ Connect.
    - Enter **DBA** as the User ID
    - Enter **SQL** as the Password
    - Click the Database tab, and in the Database File box, enter the full path and file name of the new database file.
    - Click OK.

6. Use the read statement. Remember that this statement demands that you enclose the file name in double quotes.
    - Execute the SQL statement:

      ```
      READ "c:\\My Scripts\\newdemo.SQL"
      ```

You can use these basic steps to modify other databases.

Summary

This tutorial introduced only some of the basic features of PowerDesigner. In fact, it is capable of handling the complete design or modification of your database schema, including all tables, views, indexes, references, triggers, and procedures.

Domains

Other features greatly simplify the task of designing larger databases. For example, you can specify specific **domains**. A domain holds a particular type of data, such as a phone number. It has a data type associated with it, but is more specific. For example, you can create a domain of identification numbers. Whenever you need an identification number in a table, you can associate that column with the identification number domain. All properties and checks associated with that domain are attached automatically.

Domains reduce repetitive definitions. In doing so, they not only reduce your work, but also reduce the chance that you will erroneously use a different type definition or check procedure. Rather than identify a column as simply an integer, you specify what specific type of data that column contains. All instances of that data type share a common definition.

Business rules

A business rule is a written expression of the way a business operates. For example, *the order shipped date must be greater than or equal to the order date* is a business rule.

Business rules fall into four categories:

- ♦ **Definition** Expresses inherent properties of an object. Definitions typically describe entities.
- ♦ **Fact** Expresses certainty or existence. Facts typically describe relationships.
- ♦ **Validation** A constraint on a value.
- ♦ **Formula** Calculation used to produce values.

Business rules are particularly handy because they relate directly to the task that a customer requires that a database perform. By recording business rules and attaching them to particular objects, you can ensure that a database performs the required tasks.

Where do I go from here?

☞ For more information about PowerDesigner, click Help in the toolbar of the PowerDesigner main window to access three books: *PowerDesigner General Features Guide*, *PowerDesigner PDM User's Guide*, and *PowerDesigner Report User's Guide.*

☞ For more PowerDesigner tutorials, click Start ➤ Programs ➤ Sybase SQL Anywhere 9 ➤ PowerDesigner 7 ➤ PDM Getting Started. This accesses the book *Physical Data Model Getting Started.*

☞ For more information about database design, see “Designing Your Database” [*ASA SQL User’s Guide*, page 3].

CHAPTER 7

Tutorial: Synchronizing Databases with MobiLink

About this chapter

This chapter provides a tutorial to guide you through the process of synchronizing two Adaptive Server Anywhere databases via MobiLink. One of these databases is the consolidated database, and the other is a remote database. In this tutorial, you create these databases and then synchronize the two.

Contents

About MobiLink

Data replication is the sharing of data among physically distinct databases. Sybase provides three distinct technologies for data replication:

♦ MobiLink

♦ SQL Remote

♦ Replication Server

MobiLink synchronization enables replication between an ODBC-compliant consolidated database and Adaptive Server Anywhere or UltraLite remote databases. In this tutorial an Adaptive Server Anywhere remote database is used. The consolidated database can be made by Sybase Adaptive Server Anywhere, Sybase Adaptive Server Enterprise, Oracle, Microsoft SQL Server, or IBM DB2.

MobiLink is designed for synchronization involving a consolidated data server and large numbers of remote databases, typically including many mobile databases. Administration and resource requirements at the remote sites are minimal. The system is connection-based and a remote site can connect as often as desired. At the end of each connection, the databases are fully synchronized.

MobiLink works by lumping the results of multiple transactions on the remote database into one set of changes applied to the consolidated database. Since synchronization always occurs at a transaction boundary, referential integrity is preserved. The order of the individual changes made during the component transactions is not preserved: since uncommitted data is never replicated, data integrity is preserved.

☞ For more information about synchronization strategies, including complete MobiLink documentation, see the “About This Manual” [*MobiLink Synchronization User’s Guide*, page **??**].

☞ For an introduction to SQL Remote synchronization, see “Tutorial: Replicating Data with SQL Remote” on page 93.

Introduction

In this tutorial, you create a consolidated database and a remote database. You write synchronization publications and subscriptions. You then synchronize these databases using MobiLink synchronization technology.

Requirements

The tutorial requirements are given below.

Requirement	Discussion
Timing	The tutorial should take 50 minutes.
Software	♦ A full Adaptive Server Anywhere 9 installation. ♦ A full installation of MobiLink synchronization server.
Competencies and experience	Knowledge and/or experience with command processing. Competent at connecting to your database using ODBC and Sybase Central. For more information on ODBC, see "Lesson 4: Create an ODBC data source" on page 42.

Goals

The goals for the tutorial are to gain competence and familiarity with:

- The MobiLink synchronization server and client as an integrated system
- Executing MobiLink synchronization server and client commands
- The MobiLink synchronization server and client commands and options.

Key concepts

The MobiLink synchronization server connects to the consolidated database using the ODBC interface. The MobiLink synchronization client connects to your remote database. The MobiLink synchronization server and client function as a pair, managing the upload and download of data from one database to another.

- Important concepts you will learn in this tutorial include:
- MobiLink synchronization server, MobiLink synchronization client
- ODBC connection, ODBC data source, synchronization subscription and publication, consolidated server, remote databases.

Lesson 1: Create your databases

MobiLink synchronization requires that you have compatible consolidated and remote databases, data in database tables, and ODBC data sources for each database.

Tip
All command work in the tutorial is assumed to be taking place in the directory you make called *C:\MLTutorial*.

Create your database files

The first step is to create each of the databases. In this procedure, you build a consolidated database and a remote database using the *dbinit* executable from the command prompt.

Tip
Creating a database file using *dbinit* is similar to formatting a disk — you have a database file with no user tables or procedures. You create your database schema when you define, within the newly initialized file, various user-defined tables and procedures.

For more information on the dbinit utility, see “Creating a database using the dbinit command-line utility” [*ASA Database Administration Guide*, page 486].

❖ To create your database files

1. At a command prompt, go to the directory you made called *Samples\MobiLink\introducing_ML_tutorial* directory.

2. Create a file for your consolidated database. At a command prompt type:

   ```
   dbinit consol.db
   ```

3. Create a file for your remote database. At a command prompt type:

   ```
   dbinit remote.db
   ```

4. Verify the successful creation of these database files by typing the following at a command prompt:

   ```
   dir
   ```

5. Your database files should appear in the directory listing. If not, review your procedures and repeat Steps 1 or 2 as needed.

Create ODBC data sources

☞ You are now ready to build ODBC data sources through which you can connect to your Adaptive Server Anywhere databases.

☞ For more information on creating ODBC data sources see “The Data Source utility” [*ASA Database Administration Guide*, page 472].

❖ To create ODBC data sources

1. Create your ODBC data source for a consolidated database by typing the following from a command prompt:

   ```
   dbdsn -w test_consol -y -c "uid=DBA;pwd=SQL;dbf=C:\
            MLTutorial\consol.db;eng=Consol"
   ```

2. Create an ODBC data source for a remote database by typing the following from a command prompt:

   ```
   dbdsn -w test_remote -y -c "uid=DBA;pwd=SQL;dbf=C:\
            MLTutorial\remote.db;eng=remote"
   ```

Now you can verify your data sources.

❖ To verify your new data sources

1. Start the ODBC Administrator:

   - Choose Start ➤ Programs ➤ SQL Anywhere 9 ➤ Adaptive Server Anywhere ➤ ODBC Administrator.

   The ODBC Data Source Administrator appears.

2. Click the User DSN tab.

3. Scroll through the list to find your new data sources.

4. Select a data source and click Configure.

   The ODBC Configuration for Adaptive Server Anywhere dialog opens.

5. On the ODBC tab, test your data source by clicking the Test Connection button.

The MobiLink synchronization server and the MobiLink synchronization client use the ODBC data sources to connect to the consolidated and remote databases, respectively.

Populate your databases

You can now create tables for each of your newly initialized databases by executing SQL statements in scripts using Interactive SQL. The scripts contain SQL statements that create tables in the consolidated and remote databases and insert data. The scripts also create synchronization subscriptions and publications on the remote.

❖ To run scripts from Interactive SQL

1. Start Interactive SQL.
   - From the Start menu, choose Programs ➤ SQL Anywhere 9 ➤ Adaptive Server Anywhere ➤ Interactive SQL

     or
   - From a command prompt, type *dbisql*.

2. Connect to the consolidated database.

3. Create a table in the consolidated database and add some rows to the table:
   - Enter the following instructions:

```
CREATE TABLE cust (
   cust_id int default autoincrement primary key,
   emp_id int,
   cust_name varchar( 128 )
   );
-- add data to cust table
INSERT INTO cust ( emp_id, cust_name ) VALUES ( 1,
         'cust1' );
INSERT INTO cust ( emp_id, cust_name ) VALUES ( 1,
         'cust2' );
INSERT INTO cust ( emp_id, cust_name ) VALUES ( 2,
         'cust3' );
COMMIT;
```

4. Verify the successful creation of the table:
   - Enter the following command and verify that three rows are returned:

```
SELECT * FROM cust
```

5. Now repeat steps 2, 3, and 4 for the remote database *remote.db*, using the following SQL statements. No rows are added to the table in the remote database.

```
CREATE TABLE cust (cust_id int default autoincrement primary
         key,
   emp_id int,
   cust_name varchar( 128 )
   )
```

   The query SELECT * FROM cust should display no rows.

Tip

If you need to start your database at any time, the following commands, for the consolidated and remote databases, can be used from the *C:\MLTutorial* directory: `dbeng9 consol.db dbeng9 remote.db`

Lesson 2: Prepare the databases for synchronization

Synchronization is governed by the following:

- **Synchronization publications, users, and subscriptions** These are defined in each remote database.

- **Synchronization scripts** These are written in SQL and held in the consolidated database. Alternatively, you can write synchronization scripts in Java and store them in a location accessible by the MobiLink synchronization server. In this tutorial we use SQL scripts.

You can write, view and modify synchronization scripts as well as publications and subscriptions using Sybase Central.

Create a synchronization subscription and publication

The MobiLink synchronization publication, user, and subscription are necessary for MobiLink synchronization to happen. Each is defined in the remote database.

❖ To add a publication and synchronization subscription to the remote database

1. Start Sybase Central and connect to your remote database:
   - From the Start Button, click Programs ➤ SQL Anywhere 9 ➤ Sybase Central.
   - Right-click Adaptive Server Anywhere 9 and choose Connect from the popup menu.
   - Enter an ODBC data source name of test_remote and click OK to connect.

2. Add a publication to the remote database:
   - In Sybase Central, open the remote database.
   - Open the Publications folder.
   - From the File menu, choose New ➤ Publication. The Publication Creation wizard appears.
   - Name the publication Customer and click Next.
   - Double-click the table **cust** to add it to the list of selected tables and click Finish to create the publication.

3. Add a MobiLink user name to the remote database:
   - In Sybase Central, open the remote database.
   - Open the MobiLink Users folder.

- ♦ From the File menu, choose New ➤ MobiLink User. The MobiLink User Creation wizard appears.
- ♦ Name the user **ml_user** and click Finish to create the user.

4. Subscribe the MobiLink user to the publication:
   - ♦ In the left pane, open the Publications folder so that the Customer publication is visible.
   - ♦ In the right pane, click the ml_user MobiLink user.
   - ♦ Drag the MobiLink user onto the Customer publication.
   - ♦ You are asked whether to subscribe the user to the publication. Click Yes.

5. Add address information to the subscription:

   Address information is used to enable the MobiLink client to locate the correct MobiLink synchronization server.
   - ♦ Right click the MobiLink user ml_user, and choose Properties from the popup menu.
   - ♦ On the Connection tab, enter the following information, leaving the other fields blank:
     - **Host** localhost.
       If the MobiLink synchronization server were running on a different machine, you would enter the machine name or IP number instead of **localhost**.
   - ♦ Click OK to add the address information.

This completes the preparation of the remote database. The next step is to add synchronization scripts for the consolidated database.

Add synchronization scripts to the consolidated database

Each script belongs to a designated **script version**. You must add a script version to the consolidated database before you add scripts.

❖ To add a script version

1. Start Sybase Central and connect to the test_consol data source using the MobiLink plug-in.
2. Open the Versions folder.
3. Double click Add Version. Name the new version **default**.

❖ To add synchronized tables to your consolidated database

1. Open the Tables folder beneath the MobiLink Synchronization9 icon.
2. Open the DBA container. Right-click the cust table and add it to synchronized tables.

Now that you have designated these tables as tables to be synchronized, you can add a new table script for each upload and download to the consolidated database.

❖ To add table scripts to each synchronized table

1. Open the Synchronized Tables folder, and double-click the cust table.
2. Double click Add Table Script in the right pane. The Add Synchronizing Table Script wizard appears.
3. Select the **upload_insert** event from the drop list.
4. Click Finish.
5. Type the following code into the dialog:

```
INSERT INTO cust ( cust_id, emp_id, cust_name )
VALUES ( ?, ?, ? )
```

6. Save the script and close the dialog.
7. Repeat this step for the download_cursor event, using the following script:

```
SELECT cust_id, emp_id, cust_name
FROM cust
```

You have now generated scripts that perform a snapshot synchronization of your database, uploading new rows to the consolidated database and downloading rows from the consolidated database to the remote. In a complete MobiLink installation, you would add scripts for other events.

Lesson 3: Start the MobiLink synchronization server

☞ In this lesson you start the MobiLink synchronization server so that you can synchronize the data in the consolidated and remote database.

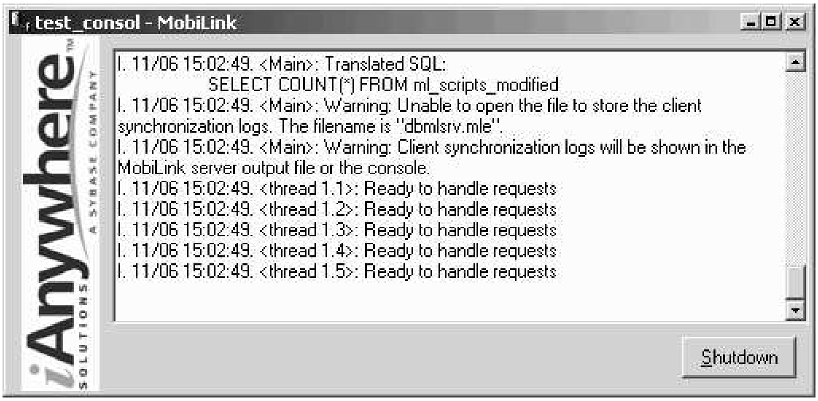
❖ To start the MobiLink synchronization server

1. From the command prompt type:

```
dbmlsrv9 -c "dsn=test_consol" -o mlserver.mls -v+ -dl -zu+
```

☞ For a detailed explanation of the meaning of the options, see “MobiLink Synchronization Server Options” [*MobiLink Synchronization Reference*, page 3].

Once you have executed the MobiLink synchronization server command, the output below appears.



```
I. 11/06 15:02:49. <Main>: Translated SQL:
               SELECT COUNT(*) FROM ml_scripts_modified
I. 11/06 15:02:49. <Main>: Warning: Unable to open the file to store the client
synchronization logs. The filename is "dbmlsrv.mle".
I. 11/06 15:02:49. <Main>: Warning: Client synchronization logs will be shown in the
MobiLink server output file or the console.
I. 11/06 15:02:49. <thread 1.1>: Ready to handle requests
I. 11/06 15:02:49. <thread 1.2>: Ready to handle requests
I. 11/06 15:02:49. <thread 1.3>: Ready to handle requests
I. 11/06 15:02:49. <thread 1.4>: Ready to handle requests
I. 11/06 15:02:49. <thread 1.5>: Ready to handle requests
```

You can check to see that this screen appears to ensure you are ready to proceed to the next lesson in the tutorial.

Lesson 4: Run the MobiLink synchronization client utility

In this lesson you start the MobiLink synchronization client. You specify connection parameters at the command prompt using the `-c` option with the dbmlsync client utility. These parameters are for the remote database.

❖ To run the MobiLink synchronization client

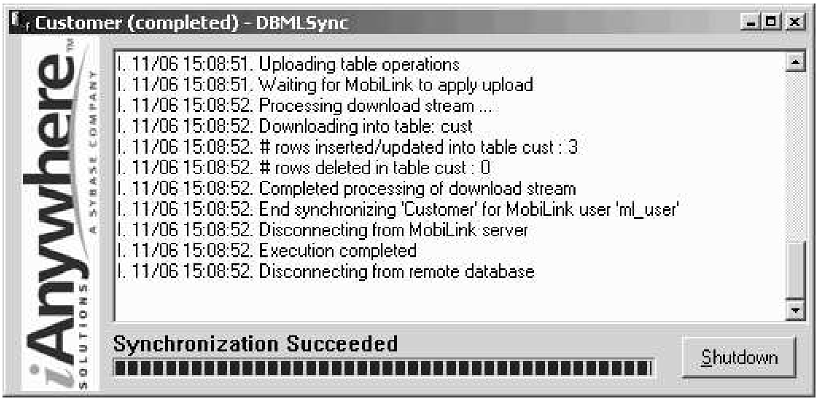
1. Ensure the MobiLink synchronization server is started.

2. At the command prompt type:

   ```
   dbmlsync -c "dsn=test_remote" -o dbmlsync.out -v
   ```

   ☞ For more information see “MobiLink synchronization client” [*MobiLink Synchronization Reference*, page 36].

Once you have executed the MobiLink synchronization remote database command, the output below appears.


Customer (completed) - DBMLSync
iAnywhere™ SOLUTIONS A SYBASE COMPANY
Shutdown

```
I. 11/06 15:08:51. Uploading table operations
I. 11/06 15:08:51. Waiting for MobiLink to apply upload
I. 11/06 15:08:52. Processing download stream ...
I. 11/06 15:08:52. Downloading into table: cust
I. 11/06 15:08:52. # rows inserted/updated into table cust : 3
I. 11/06 15:08:52. # rows deleted in table cust : 0
I. 11/06 15:08:52. Completed processing of download stream
I. 11/06 15:08:52. End synchronizing 'Customer' for MobiLink user 'ml_user'
I. 11/06 15:08:52. Disconnecting from MobiLink server
I. 11/06 15:08:52. Execution completed
I. 11/06 15:08:52. Disconnecting from remote database
```

Synchronization Succeeded

You can check to see that this screen appears to ensure your synchronization has succeeded. You can then connect to the remote database and confirm that the cust table has three rows in. These rows were added at the consolidated database.

You may wish to add more rows at the remote and consolidated database, and synchronize them.

Clean up

Be sure to delete all tutorial-related data sources and databases once you have finished this tutorial.

Summary

Now that you have two databases and have synchronized their contents, you can do so any time by running the synchronization server and client.

During this tutorial, you accomplished the following tasks.

- Created new Adaptive Server Anywhere databases to serve as consolidated and remote databases
- Wrote synchronization publication and subscription definitions
- Created scripts to control data upload and download.
- Started a MobiLink synchronization server, the MobiLink synchronization client, and synchronized the remote database with the consolidated database

Learning accomplishments

In this tutorial you:

- Gained familiarity with the MobiLink synchronization server and client as an integrated system and learned MobiLink synchronization server and client commands and options
- Acquired competence in executing MobiLink synchronization server and client commands
- Became competent at writing synchronization scripts.

What's next?

You may need to learn more about MobiLink functioning to get the most out of MobiLink. The following areas are good starting points for further reading:

- Try the next tutorial in our series "Introduction" [*MobiLink Synchronization User's Guide,* page 370].
- Read the introductory chapter on MobiLink "Introducing MobiLink Synchronization" [*MobiLink Synchronization User's Guide,* page 3].
- Read the chapter on running the MobiLink synchronization server "Running MobiLink Outside the Current Session" [*MobiLink Synchronization User's Guide,* page 329].
- Get to know the MobiLink client utility "MobiLink synchronization client" [*MobiLink Synchronization Reference,* page 36].

CHAPTER 8

Tutorial: Replicating Data with SQL Remote

About this chapter

This chapter shows you how to set up a simple replication system using SQL Remote.

Contents

About SQL Remote

Data replication is the sharing of data among physically distinct databases. Changes made to shared data in any one database are replicated in the other databases. Sybase provides three distinct strategies for data replication:

- SQL Remote
- MobiLink
- Replication Server

SQL Remote is designed for two-way replication involving a consolidated data server and a very large number of remote databases, typically including many mobile databases. Administration and resource requirements at the remote sites are minimal. This system is message-based.

In a SQL Remote installation, the central database must be either Sybase Adaptive Server Anywhere or Sybase Adaptive Server Enterprise.

SQL Remote replicates data by scanning the transaction log and preparing messages, as appropriate, for each transaction. It orders these messages and sends them to the remote or consolidated site. When processing received messages, SQL Remote always processes them in the same order as they were applied to the other database. When necessary, it automatically delays processing a message until all earlier messages have been applied.

☞ For more information about synchronization strategies, including complete SQL Remote documentation, see the “About This Manual” [*SQL Remote User’s Guide*, page **??**].

☞ For an introduction to MobiLink synchronization, see “Tutorial: Synchronizing Databases with MobiLink” on page 81.

Lesson 1: Getting Started

This tutorial describes how to set up a simple SQL Remote replication system.

With SQL Remote, you can carry out two-way replication between a central database (called the **consolidated database**) and a set of **remote databases**. These remote databases may be on laptop computers, while the consolidated database may be on a network in an office. All the setup and administration is carried out at the consolidated database.

This tutorial describes a very simple case, replicating one table from the sample database to a single remote database on the same machine.

In the tutorial, you act as the system administrator of a consolidated Adaptive Server Anywhere database. The replication system consists of a simple table from the sample database. The table we replicate is the department table, which is one of the simplest in the database.

The tutorial takes you through the following steps:

1. Setting up the consolidated database.
2. Creating a file-sharing replication system with a single remote database.
3. Replicating data between the two databases.

Create directories for the tutorial

Before you start, you need to create directories to hold the databases and other files you create during the tutorial. You should also create a copy of the sample database in case you need to recreate it in its original form.

❖ Prepare for the tutorial

1. Create a directory to hold the files you make during this tutorial, for example, *c:\tutorial*.
2. Create two subdirectories of *tutorial* called *DBA* and *field*, for example, *c:\tutorial\DBA* and *c:\tutorial\field*. These subdirectories hold messages for each of the two user IDs in the replication system.
3. Create a copy of the sample database. Create a new folder for the copy, for example, *demoback*, and copy *asademo.db* into it.

What next?

Now you're ready to create the consolidated database.

Lesson 2: Set up the consolidated database

This section shows you how to prepare the consolidated database of a simple SQL Remote replication system. You require DBA authority to do this.

To prepare a consolidated database for replication, this lesson takes you through the following steps:

1. Create a message type to use for replication.
2. Grant PUBLISH permissions to a user ID to identify the source of outgoing messages.
3. Grant REMOTE permissions to all user IDs that are to receive messages.
4. Create a publication describing the data to be replicated.
5. Create subscriptions describing who is to receive the publication.

Add a SQL Remote message type

All messages sent as part of replication use a message type. A message type description has two parts:

- A message link supported by SQL Remote. In this tutorial, we use the FILE link. This is a file sharing method, where the messages are left in a file on disk, and read by the intended recipient. SQL Remote also supports ftp and e-mail protocols.
- An address for this message link, to identify the source of outgoing messages. In this tutorial we use a file path to specify where the messages will be left.

Message types are created in all new databases, but you do need to supply an address for the message type you will use.

❖ Add an address to a message type

1. From Sybase Central, connect to the sample database as user ID **DBA** using the password **SQL**, and open the sample database container (**asademo**) in the left pane.
2. Click the SQL Remote Users folder in the left pane.
3. Click the Message Types tab in the right pane.
4. Right-click the FILE message type, and choose Properties.

5. Enter a publisher address to provide a return address for remote users. The publisher address is the directory you created in Lesson 1 to hold messages for the consolidated database (*DBA*). For example, *c:\tutorial\DBA*.

6. Click OK to save the message type.

Add the publisher to the database

Each database in a SQL Remote replication system needs a single user ID that identifies the **publisher** of the data. Here, we make the DBA user ID the publisher.

❖ Set the publisher

1. Click the Users & Groups folder in the left pane.

2. Right-click DBA in the right pane, and choose Change to Publisher from the popup menu.

   Publisher appears in the Type column beside DBA.

A database can have only one publisher. You can find out who the publisher is at any time by opening the Users & Groups folder.

Add a remote user to the database

Each remote database is identified in the consolidated database by a user ID with REMOTE permissions.

When a remote user is added to a database, the message system they use and their address under that message system need to be stored along with their database user ID.

❖ Add a remote user

1. Open the SQL Remote Users folder in the left pane.

2. Choose File ➤ New ➤ SQL Remote User.

   The Remote User Creation wizard appears.

3. Type the name **field** as the name of the new remote user, and click Next.

4. Ensure that the user is allowed to connect, and type the password **field**. Confirm the password by entering it again. Click Next.

5. Select DBA permissions as well as REMOTE DBA permissions for the remote user.

6. Select the File message type and enter the remote address **field** in the text box. Click Next.

7. Select Send Then Close. Click Next.

8. Click Finish to create the remote user.

   The remote user field appears in the SQL Remote Users folder.

Notes

If you forget to set DBA permission in the wizard, you can set it by double-clicking the user, and checking DBA on the Authorities tab.

You have now created the users who will use this system.

Add publications and subscriptions

This section describes how to add a publication to a database, and how to add a subscription to that publication for a user. The publication replicates all rows of the table department.

❖ Add a publication

1. Open the Publications folder.

2. From the File menu, choose New ➤ Publication.

   The Publication Creation wizard appears.

3. Name the publication DepartmentPub on the first page of the wizard. Click Next.

4. Select department from the list of Available Tables. Click Add.

   The table appears in the list of Selected Tables on the right.

5. Click Finish to create the publication.

Add a subscription

Each user ID that is to receive changes to a publication must have a **subscription** to that publication. Subscriptions can only be created for a valid remote user. You need to add a subscription to the DepartmentPub publication for the remote database user field.

❖ Add a subscription

1. Open the Publications folder.
2. Click the SQL Remote Subscriptions tab in the right pane.
3. From the File menu, choose New ➤ SQL Remote Subscription.
   The Create SQL Remote Subscription for Remote User dialog appears.
4. Choose to create a submscription for the user field. Click Next
5. Click Finish.

You have now set up the consolidated database.

What next?

You can now create the remote database.

Lesson 3: Set up the remote database

The remote database needs to be created and configured in order to send and receive messages and participate in a SQL Remote setup.

The database extraction utility enables you to carry out all the steps needed to create a remote database complete with subscriptions and required user IDs.

Create the remote database

You need to extract a database from the consolidated database for remote user field.

❖ Extract the remote database

1. Right-click the sample database asademo (DBA), and choose Extract Database from the popup menu.

   The Extract Database wizard appears.

2. Click Next on the introductory page of the wizard.

3. Choose to extract the asademo database. Click Next.

4. Choose to extract at isolation level 3. Click Next.

5. Choose to Start Subscriptions Automatically for user field. Click Next.

6. Enter a location for the SQL file that will build the database. For example, choose *c:\tutorial\field.sql.* As well, choose to extract the structure and data (the default setting).

7. Choose to extract all parts of the schema and leave the maximum view dependency at its default setting. Click Next.

8. Choose not to Extract Fully Qualified Publication Definitions. Click Next.

9. Leave the location to save the data at its default value.

10. Choose Create and Reload Into the Following New Database, and enter a path and filename for the database. For example, *c:\tutorial\field.db.* Click Finish.

    The Extracting Database window appears, displaying the progress of the extraction. When completed, close the window.

Note

In a proper SQL Remote setup, the remote database **field** would be loaded on to the computer, together with a database server and any client applications required. For this tutorial, we leave the database where it is and use Interactive SQL to input and replicate data.

Verify that the database is created properly

To see what the extraction utility has done, connect to the *field* database and confirm that all the database objects are created.

❖ Browse through the remote database

1. In Sybase Central, click the Connect button.

   The Connect dialog appears.

2. On the Identification tab, enter the user ID **field** and password **field**.

3. On the Database tab, enter the database path (for example, *c:\tutorial\field.db*). Click OK to connect to the database.

   The database field appears in the left pane of Sybase Central.

4. Open the Tables folder. The department table, owned by user DBA, is in the list.

5. In the left pane, click the Department table. In the right pane, click the Data tab to show the five rows of the department table.

6. Open the SQL Remote Users folder. The field user appears in the right pane, showing that field is the publisher for this database. Any data sent from this database comes from the user field, just as any data from the consolidated database comes from the user DBA.

   You will see that user ID DBA is present, and is designated as a consolidated user. This means DBA is the publisher of the consolidated database (above the field database in the hierarchy).

7. Open the Publications folder. You will see that the DepartmentPub publication is present.

What next?

The system is now ready for replication.

Leave Sybase Central open for use later in the tutorial.

Lesson 4: Replicate data

You now have a replication system in place. In this section, data is replicated from the consolidated database to the remote database, and from the remote to the consolidated database.

Add data to the consolidated database

First, enter a row in the consolidated database.

❖ Enter data in the consolidated database

1. Add a row into the department table.
   - In Sybase Central, open the Tables folder of the consolidated database (asademo), select the department table and click the Data tab in the right pane.
   - Click the plus sign in the toolbar to add a row. Enter the following values:

Column	**Value**
dept_id	202
dept_name	Eastern Sales
dept_head_id	Leave as (NULL)

   - Press Enter. If a confirmation dialog appears, click Yes to update the row.

   The next step is to send the new row to the remote database.

To send data to the remote database, you run the Message Agent at the consolidated database. The *dbremote* program is the Message Agent for Adaptive Server Anywhere.

❖ Send the data to the remote database

1. At a command prompt, navigate to your tutorial directory. For example, navigate to *c:\tutorial*.

2. To run the Message Agent against the consolidated database, execute the following command:

   ```
   dbremote -c "dbn=asademo;uid=DBA;pwd=SQL"
   ```

   The SQL Remote window appears, and displays messages about the status of replication.

3. When SQL Remote displays the message `Execution completed`, click Shutdown.

   To receive the insert statement at the remote database, you must run the Message Agent, *dbremote*, at the remote database.

❖ Receive data at the remote database

1. At a command prompt, change to your tutorial directory. For example, navigate to *c:\tutorial*.

2. Execute the following command to run the Message Agent against the **field** database:

   ```
   dbremote -c "dbn=field;uid=field;pwd=field"
   ```

   The SQL Remote window appears, informing you of the status of replication. The window indicates that a message was received from DBA.

3. When SQL Remote displays the message `Execution completed`, click Shutdown.

Notes

♦ The SQL Remote window displays status information while running. This information can be output to a log file for record keeping. You will see that the Message Agent first receives a message from *asademo*, and then sends a message. This return message contains confirmation of successful receipt of the replication update; such confirmations are part of the SQL Remote message tracking system that ensures message delivery even in the event of message system errors.

♦ Depending on the current status of the database, there are three different connectivity parameters you can use to connect to a database:

- **dbf** Connect to the database using the database file. This parameter requires you to specify the database file itself. If no server is currently running, a server will be started and the database will be loaded onto it. If a server is already running, the database will be loaded onto the default server.
- **dbn** Connect to the database using the database name. This parameter requires you to specify the name of the database. You can only use this parameter when the database is already running.
- **dbs** Connect to the database using the database source. A data source is a collection of parameters stored in the system registry or in a set of files. The source is referenced simply by its name.

  ☞ For more information about data sources, see “DataSourceName connection parameter [DSN]” [*ASA Database Administration Guide*, page 186].

Verify that the data has arrived

From Sybase Central, inspect the department table to verify that the row has been received.

❖ Verify that the data has arrived

1. Open the *field* database folder.
2. In the left pane, select the department table. In the right pane, click the Data tab to view the rows in the table.

   You will see that the department table contains the Easter Sales department entered at the consolidated database.

Replicate from the remote database to the consolidated database

Now try entering data at the remote database and sending it to the consolidated database. Only the outlines are presented here.

❖ Replicate data from the remote database to the consolidated database

1. Insert a row at the remote database, for a department with a dept_id of 203 and a dept_name of Western Sales.
2. Run *dbremote* to send the message to the consolidated database:

   ```
   dbremote -c "dbn=field;uid=field;pwd=field"
   ```

3. Run *dbremote* to receive the message at the consolidated database:

   ```
   dbremote -c "dbn=asademo;uid=DBA;pwd=SQL"
   ```

4. Display the data in the department table at the consolidated database: It contains the Western Sales row.

What next?

The tutorial is now complete, but you should continue to the next section to restore the sample database to its original form.

Lesson 5: Restore the database and database settings

Once you have completed the tutorial it is important to undo any changes you have made to the sample database. Make sure that the following steps are completed in order to ensure that the settings are reset properly.

❖ Delete the remote user

1. In the left pane of Sybase Central, click the Users & Groups folder for the sample database.

2. Right-click the field user, and choose Delete from the popup menu. Click Yes to remove the user from the list.

❖ Delete the publication

1. Click the Publications folder.

2. Right-click DepartmentPub in the left pane and choose Delete from the popup menu. Click Yes to confirm the deletion.

❖ Revoke the publishing status from the sample database

1. Click the Users and Groups folder in the left pane.

2. Right-click the DBA user in the left pane, and select Revoke Publisher from the popup menu.

Revoking a status while running the database
The status of a user cannot be modified while the user is running the database. Make sure that the user is idle before revoking or invoking a status.

❖ Restore the original message type settings

1. Select the SQL Remote Users folder and click the Message Types tab in the right pane.

2. Right-click the File message type in the right pane and choose Properties.

3. Delete the publisher address and click OK to restore the settings.

❖ Delete the remote database

1. In the left pane, select the Adaptive Server Anywhere 9 plug-in.
2. In the right pane, click the Utilities tab.
3. In the right pane, double-click Erase Database.

   The Erase Database wizard appears.
4. Follow the instructions in the wizard.
5. Enter the name of the database you want to erase or click Browse to search for the database. For example, enter *c:\tutorial\field.db.* Click Finish to erase the database file.

Notes

Now that the tutorial is complete, it might be a good idea to delete the tutorial directory (*c:\tutorial)* in order to save space.

Restore the data in the database

The most important part of the cleanup process is to ensure that the changes to the sample database are reversed. The integrity of the sample database is very important in order to carry out other tutorials in later chapters of the manual.

❖ Delete the inserted data from the sample database

1. From Sybase Central, display the data in the department table of the asademo database.
2. Delete the rows with dept_id values of 202 and 203 to restore the table to its original state.

Summary

In this tutorial you learned how to

- Prepare the consolidated database of a simple replication system.
- Create and configure a remote database.
- Replicate data in both directions between the two databases.
- Restore your database and database settings.
- Verify all the steps.

Where do I go from here?

☞ For more information about SQL Remote, see the “About This Manual” [*SQL Remote User’s Guide*, page **??**].

☞ For an introduction to MobiLink synchronization, see “Tutorial: Synchronizing Databases with MobiLink” on page 81.

CHAPTER 9

Tutorial: Creating Reports with InfoMaker

About this chapter

This chapter includes a brief tutorial of InfoMaker report making that gets you started and teaches you the basic skills required to be productive in the InfoMaker environment.

Contents

About InfoMaker

InfoMaker is a powerful reporting and data maintenance tool. With InfoMaker, you can create the following objects:

- ♦ Reports to view data.
- ♦ Forms to view and change data.
- ♦ Queries to automatically retrieve data for reports or forms.
- ♦ Pipelines to pipe data from one database (or DBMS) to another.
- ♦ Applications to bundle reports and forms and distribute them to users.

InfoMaker provides built-in connectivity to a broad range of desktop and server-based databases. When you work in InfoMaker, you work in a graphical environment—and working with data in this environment means you don't need to understand SQL. InfoMaker creates all SQL statements behind the scenes as you build your reports and other objects graphically.

☞ For more information about InfoMaker, choose Start ➤ Programs ➤ Sybase SQL Anywhere 9 ➤ InfoMaker 8.0 ➤ Online Help Files. Complete documentation is also available within the application by clicking the Help menu.

Lesson 1: Getting Started

To use this InfoMaker tutorial, you must be connected to the Adaptive Server Anywhere sample database. To connect to the SQL Anywhere Studio 8.0 sample database you need to create a Database Profile.

❖ To start InfoMaker

1. From the Start menu, choose Programs ➤ Sybase SQL Anywhere 9 ➤ InfoMaker 8.0 ➤ InfoMaker.

   The InfoMaker initial window appears. It includes the PowerBar, which has buttons for accessing the InfoMaker painters and online Help.

❖ To create a Database Profile for the SQL Anywhere Studio sample database with InfoMaker

1. In InfoMaker, open the Database Profiles dialog by clicking DB Profile in the PowerBar.

2. Select ODBC and click New.

3. The Database Profile Setup dialog appears.
   - Enter the Profile Name **Anywhere**.
   - Select ASA 9.0 Sample as the Data Source.
   - Enter **DBA** as the user ID.
   - Enter **SQL** as the password.
   - Click OK.

   The profile Anywhere should now be listed under ODBC.

❖ To connect to the Adaptive Server Anywhere sample database once a Data Source has been created

1. If it is not already open, open the Database Profiles window by clicking DB Profile in the PowerBar.

2. Select Anywhere from the ODBC group and click Connect.

What next?

Next you will create a table using an InfoMaker wizard.

Lesson 2: Create a basic report

This section shows you how to create a basic report, how to preview and save it, and how to change the settings in your design environment.

Create a report

There are many types of reports you can create. This section shows you how to create a table.

❖ To create the report

1. Click the New button.
2. In the New dialog, select the Object tab.
3. Double-click the icon labeled Tabular. This selects the tabular presentation style.

   The wizard for creating tabular reports appears.
4. Select the Quick Select data source, and ensure that Retrieve on Preview is selected. Click Next.

   The Quick Select dialog appears. This allows you to select a database table and some or all of the table's columns.
5. Select the table called contact.
6. Select the following columns: last_name, first_name, title, phone, and fax. (You may need to scroll down.)

   InfoMaker moves the columns you selected to the grid at the bottom of the dialog. You can use this grid for reordering the columns, and for providing sort and selection criteria.
7. Click OK.

   The Select Color and Border Settings dialog appears.
8. You can use the defaults, so click Next.

   A dialog summarizing your specifications appears.
9. Click Finish.

Preview your report

In this section you will view your report to see what it looks like before printing it.

The main InfoMaker window has toolbars in the top section. Underneath the toolbars is the preview pane, which occupies the rest of the window. At the top of the preview pane is a title bar. At the left of the title bar is the name of the report you are previewing, and at the right are Minimize/Maximize buttons for adjusting the size of the preview pane.

❖ To preview your report

1. Position your pointer in the upper right corner of the preview pane and click the Maximize button.

   The Preview pane fills the entire window. Notice that it now contains the header information for the report, as well as information about the database.

2. To see how to turn off the rulers, click in the section that contains the data, and choose File ➤ Print Preview Rulers.

3. Click the Minimize button in the upper right corner of the preview pane to return the preview pane to its original size and location.

Save the report

❖ To save the report

1. Choose File ➤ Save.

2. In the Reports box, type **contacts_by_jobrole**. This becomes the name of the report.

3. Click in the Comments box and type **This report shows my contacts grouped by job role**.

4. Press Enter.

Set up the design environment

In this section you will make modifications to the controls, grid and ruler.

❖ To set up the design environment

1. From the Design menu, choose Options.

   The Report Options property sheet appears.

2. Make sure that the following options are selected: Show Grid, Show Ruler, and Show Edges.

3. Make sure that Snap to Grid is not selected.

4. Click Help to see descriptions of all options.

5. Click OK.

What next?

You are now ready to customize the look of your report.

Lesson 3: Enhance your report

In this section you'll learn how to make a number of enhancements to your report, including

- Sorting the data.
- Creating and formatting headers and titles.
- Adding computed fields such as dates, page numbers, and totals.

Define sorting and grouping

In this section you group contacts by title. To do this, you sort the data by title and then specify grouping by title.

❖ To group contacts

1. Define the sorting. Ascending order is the default.
    - From the Rows menu, choose Sort.

      The Specify Sort Columns dialog appears.
    - Drag tile from Source Data box to the Columns box.
    - Drag last_name and first_name into the Columns box.
    - Click OK.
2. Define the grouping based on the title column, so that all contacts with the same title are grouped together:
    - From the Rows menu, choose Create Group.

      The Specify Group Columns dialog appears.
    - Drag title to the Columns box and click OK.

The grouping is completed. Expand the Preview pane in the same way you did earlier in this lesson, and you will see that the data is sorted. The grouping does not yet appear.

Enhance the report

In this section we'll enhance the report by rearranging controls, adding a title and date, adding page numbers, and adding a total.

❖ To enhance the report

1. Move the job role to the group header and change its font:
   - ♦ Move the pointer to the gray bar called 1: Header Group Title. The pointer changes to a double-arrow.
   - ♦ Drag the band down about five grid dots.
   - ♦ Drag the title column from the bottom of the preview pane into the bottom of the band for the group header.
   - ♦ With the title column still selected, click B and I on the style bar.

   Review the report in the Preview view.

2. Fix up the headers:
   - ♦ Drag the First Name and Last Name text boxes into the right side of the header band to fill the empty space.
   - ♦ Drag the first_name and last_name columns into the right side of the detail band to fill the empty space.
   - ♦ From the Edit menu, choose Select ➤ Select Text.

     This selects all the headers. It may be difficult to see that they are selected, because the band is so narrow.
   - ♦ On the stylebar, click the Left Justification button. If your headers are not already bold, click the Bold button.

   Review the report in the Preview pane.

3. Add a title to the report:
   - ♦ Drag the gray bar marked Header down about 10 grid dots.
   - ♦ From the Edit menu, choose Select ➤ Select Text. This selects all the headers.
   - ♦ Drag all the headers down close to the gray band.
   - ♦ From the Insert menu, choose Control ➤ Text.
   - ♦ Move the point of the pointer above the First Name box and down one grid dot from the top of the page. Click once. This positions the text box for the title.
   - ♦ Type **My Contacts**
   - ♦ Select size 14 from the dropdown list in the stylebar.

4. Add a date to the report:
   - ♦ From the Insert menu, choose Control ➤ Today().
   - ♦ Move the point of the pointer to the upper left corner of the report and click. This places the date into the report.

5. Add page numbers to the report:
   - ♦ Drag the gray bar marked Footer down about four grid dots. If necessary, use the scrollbar to make room at the bottom of the Design view. Dragging down the footer bar makes space for the page number in the footer band.
   - ♦ From the Insert menu, choose Control ➤ Page n of n.
   - ♦ Place the pointer about two grid dots below the center of the footer band. Click.
   - ♦ Select size 10 in the stylebar.
6. Add a total by counting the number of last names:
   - ♦ Drag the gray bar marked Summary down about six grid dots. This makes space for the total.
   - ♦ From the Insert menu, choose Control ➤ Computed Field.
   - ♦ Place the pointer about four grid dots below the center of the summary band. Click.

     The Modify Expression dialog appears.
   - ♦ In the Functions box, click Count(#x for all).
   - ♦ In the Columns box, click last_name.
   - ♦ Click Verify.

     You receive a message stating that your expression is OK. Click OK.
   - ♦ Click OK to complete the definition of the expression for the computed field.
7. Format the total:
   - ♦ From the Insert menu, choose Control ➤ Text.
   - ♦ Position the pointer to the left of the computed field and click.
   - ♦ Type **Total contacts:**
   - ♦ Drag the text box to line it up with the computed field.
   - ♦ Lasso the text and computed field by circling them with the pointer while pressing the left mouse button, and then releasing the mouse button.
   - ♦ On the Stylebar, select the font size 10, and click B and I.

❖ To print, save, and close the report

1. From the File menu, choose Print Report.
2. In the Windows Print dialog, click OK.
3. From the File menu, click Close.
4. If you are prompted to save changes, click Yes.

Summary

In this tutorial you learned how to

- Connect to InfoMaker.
- Create a basic report.
- Preview your report.
- Format the report by sorting the data, and adding headers, titles, dates, page numbers and a total.

Where do I go from here?

In addition to reports, InfoMaker provides functionality for creating graphs, queries, and data entry forms.

☞ For more information about InfoMaker, choose Start ➤ Programs ➤ Sybase SQL Anywhere 9 ➤ InfoMaker 8.0 ➤ Online Help Files. Complete documentation is also available within the application by clicking the Help menu.

Part III

Appendix

The appendix describes supported operating systems for the components of SQL Anywhere Studio.

APPENDIX A

SQL Anywhere Studio Supported Platforms

About this chapter

This chapter lists the supported platforms for the individual components of SQL Anywhere Studio.

Supported platform information is also available at *http://www.ianywhere.com/products/supported_platforms_8.html* and at *http://www.ianywhere.com/products/supported_platforms.html.*

Availability
The presence of a supported platform in this chapter does not neccessarily indicate immediate availability of the software.

Contents

Introduction

SQL Anywhere Studio provides database management and synchronization technologies for a wide variety of operating systems. It includes the following major components:

- **Adaptive Server Anywhere** A relational database system for use all the way from a Small and Medium Business (SMB) database server through desktop database applications down to Windows CE devices.
- **UltraLite** A relational database designed specifically for small devices. It runs on Palm OS and Windows CE devices.
- **MobiLink** A synchronization system that provides two-way synchronization between many Adaptive Server Anywhere or UltraLite databases and a central database server.
- **SQL Remote** A synchronization system for Adaptive Server Anywhere and Sybase Adaptive Server Enterprise databases.
- **Administration tools** A suite of tools provides management for all components in SQL Anywhere Studio.

This appendix lists which components of SQL Anywhere Studio run on which operating systems. The information is organized by component within SQL Anywhere Studio.

☞ This paper provides information current at the time of writing. For the latest information, see *http://www.ianywhere.com/products/supported_platforms.html.*

Overview

The following table summarizes platform support for the major components. The table does not list detailed information such as differences in support for individual features on the listed platforms, or which flavors and versions of UNIX are supported. For detailed information, see the tables in the remainder of this document.

The UNIX platform support is of two kinds.

- **Full edition** Complete releases of SQL Anywhere Studio are available on Sun Solaris and Linux.
- **Deployment edition** The deployment edition contains core Adaptive Server Anywhere and SQL Remote components. Some deployment releases contain the MobiLink synchronization server. Deployment edition platforms include HP-UX, IBM AIX, Compaq Tru-64, and 64-bit Linux operating systems.

☞ The supported versions of the listed operating systems are as described in more detail in “Operating system versions” on page 138.

Component	Windows (32-bit)	Windows (64-bit)	UNIX (full)	UNIX (deploy-ment)	NetWare	Windows CE	Palm OS
Adaptive Server Anywhere database server	✔	✔	✔	✔	✔	✔	
Adaptive Server Anywhere clients	✔	✔	✔	✔	✔ [1]	✔	
UltraLite databases	✔ [2]					✔	✔
UltraLite development	✔						
MobiLink synchronization server	✔	✔	✔	✔ [3]			
SQL Remote	✔	✔	✔	✔	✔	✔	
Replication Agent	✔	32-bit software	✔				
Administration tools	✔	32-bit software	✔		✔ [4]		

[1] Embedded SQL only
[2] For development and testing only
[3] AIX only
[4] Limited set

Windows and NetWare operating systems

The following tables list the supported operating systems for various components of SQL Anywhere Studio.

Unless otherwise noted, the supported versions of the listed operating systems are as described in "Operating system versions" on page 138.

Adaptive Server Anywhere

System requirements depend on the database size, workload, and required performance. The following are minimal requirements only. They apply to both the personal database server and the network database server:

- **Hardware** An Intel 486 or higher CPU, or compatible CPU, is required as a minimum. For UNIX systems, the required CPU is operating-system dependent.
- **Memory** Adaptive Server Anywhere can run with as little as 4 Mb of memory with an additional 8kb per client connection. If you use Java in the database, Adaptive Server Anywhere requires an additional 5 Mb of memory. Your computer must have this much memory in addition to the requirements for the operating system.

Component	**Windows NT/2000/XP (32-bit)**	**Windows (64-bit)**	**Windows 95/98/Me**	**Windows CE**	**NetWare**
Personal database server	✔	✔	✔		
Network database server	✔	✔	✔	✔	✔
ODBC clients	✔	✔	✔	✔ [5]	
OLE DB clients	✔	✔	✔	✔	
Embedded SQL clients	✔	✔	✔	✔	✔
Open Client clients	✔	✔	✔		
JDBC clients (jConnect)	✔	32-bit software	✔		
JDBC clients (iAnywhere JDBC driver)	✔	✔	✔		
SQL preprocessor	✔	✔	✔		
Java in the database	✔		✔		✔

[5] ODBC driver manager may not be available. May have to link directly to ODBC driver.

Component	Windows NT/2000/XP (32-bit)	Windows (64-bit)	Windows 95/98/Me	Windows CE	NetWare
Remote data access [6]	✔	✔	✔		✔
External stored procedures	✔	✔[7]	✔		✔
Dynamic cache resizing	✔	✔	✔		✔
Strong encryption	✔	✔	✔		✔
CREATE DATABASE, RESTORE DATABASE, DROP DATABASE statements	✔	✔	✔		✔
SPX protocol	✔		✔		✔

UltraLite host platforms

Component	Windows NT/2000/XP	Windows (64-bit)	Windows 98 SE	Windows CE	NetWare
SQL preprocessor (Embedded SQL only)	✔	✔	✔		
UltraLite generator (static C++ API, static Java API)	✔		✔		
UltraLite for AppForge MobileVB	✔				
UltraLite ActiveX	✔				
Native UltraLite for Java	✔				
UltraLite.NET	✔ [8]				
UltraLite for C++	✔				

☞ For information about deployment platforms, see “UltraLite target platforms” on page 136.

Development environments

You can use one of the following development environments to build UltraLite Palm applications:

[6] Requires iAnywhere Solutions ODBC driver. For a list of supported platforms, see “iAnywhere Solutions ODBC drivers supported platforms” on page 135.

[7] The older interface to external stored procedures is available only in 32-bit software.

[8] Requires Visual Studio.NET with .NET Compact Framework version 1.0.5000 or later.

- Metrowerks CodeWarrior version 8 or 9.

  ☞ See “Developing UltraLite applications with Metrowerks CodeWarrior” [*UltraLite Embedded SQL User’s Guide*, page 73].

  CodeWarrior includes a version of the Palm SDK. Depending on the particular devices you are targeting, you may want to upgrade your Palm SDK to a more recent version than that included in the development tool.

- AppForge MobileVB, using the UltraLite MobileVB component.

☞ For information on supported development tools for Palm OS targets, see “Development environments” [*UltraLite Embedded SQL User’s Guide,* page 72].

MobiLink synchronization server

Component	**Windows NT/2000/XP**	**Windows (64-bit)**	**Windows 95/98/Me**	**Windows CE**	**NetWare**
MobiLink synchronization server	✔	✔	✔		
TCP/IP synchronization	✔	✔	✔		
HTTPS synchronization	✔	✔	✔		
HTTP synchronization	✔	✔	✔		
Transport-Layer Security	✔		✔		
Java synchronization logic	✔	32-bit software	✔		
.NET synchronization logic	✔	✔	✔		

MobiLink Adaptive Server Anywhere clients

Component	**Windows NT/2000/XP**	**Windows (64-bit)**	**Windows 95/98/Me**	**Windows CE**	**NetWare**
ASA MobiLink clients (*dbmlsync*)	✔	✔	✔	✔	
TCP/IP synchronization	✔	✔	✔	✔	
HTTP synchronization	✔	✔	✔	✔	
HTTPS synchronization	✔	✔	✔	✔	
Transport-layer security	✔	✔	✔	✔	

MobiLink utilities

Component	Windows NT/2000/XP	Windows (64-bit)	Windows 95/98/Me	Windows CE	NetWare
Redirector	✔				
MobiLink extraction utility (*mlxtract*)	✔	✔	✔		

SQL Remote

SQL Remote components that operate against Adaptive Server Enterprise databases require Sybase Open Client and/or Open Server libraries.

Component	Windows NT/2000/XP	Windows (64-bit)	Windows 95/98/Me	Windows CE	NetWare
ASA Message Agent (*dbremote*)	✔	✔	✔	✔	✔
ASE Message Agent (*ssremote*)	✔	32-bit software	✔		
ASE stable queue (*ssqueue*)	✔	32-bit software	✔		
ASA extraction utility (*dbxtract*)	✔	✔	✔		
ASE extraction utility (*ssxtract*)	✔	32-bit software	✔		
File message type	✔	✔	✔	✔	✔
FTP message type	✔	✔	✔	✔	✔
MAPI message type	✔	✔	✔		
SMTP message type	✔	✔	✔	✔	✔

Replication agent

The Replication Agent requires Sybase Open Client and Open Server libraries.

Component	Windows NT/2000/XP	Windows (64-bit)	Windows 95/98/Me	Windows CE	NetWare
Replication Agent	✔	32-bit software	✔		

Administration tools

Component	Windows NT/2000/XP	Windows (64-bit)	Windows 95/98/Me	Windows CE	NetWare
Command line administration utilities	✔	✔	✔		✔
Sybase Central	✔	32-bit software	✔		
Interactive SQL	✔	32-bit software	✔		✔ [9]
DBConsole	✔	32-bit software	✔		✔ [10]

With the exception of the command-line administration utilities and the tools provided on NetWare, the administration tools employ a Java 2 runtime environment, version 1.4.1.

The following paragraphs are taken from the Java 2 SDK, Standard Edition documentation, and apply to the administration tools:

The Java™ 2 SDK is intended for use on Microsoft Windows 95, 98 (1st or 2nd edition), NT 4.0 with Service Pack 5, Me, 2000 Professional, 2000 Server, 2000 Advanced Server, or XP operating systems running on Intel hardware.

A Pentium 166MHz or faster processor with at least 32 megabytes of physical RAM is required to run graphically based applications. Forty-eight megabytes of RAM is recommended for applets running within a browser using the Java Plug-in product. Running with less memory may cause disk swapping which has a severe effect on performance. Very large programs may require more RAM for adequate performance.

You should have 70 megabytes of free disk space before attempting to install the Java 2 SDK software. If you also want to install the documentation download bundle, you will need an additional 120 megabytes of free disk space.

In non-English locales, only the Professional edition of Windows 2000 is supported.

[9] A more limited version than is provided on other operating systems

[10] A more limited version than is provided on other operating systems

UNIX and Linux operating systems

Unless otherwise noted, the supported versions of the listed operating systems are as described in "Operating system versions" on page 138.

Adaptive Server Anywhere

Component	Solaris	Linux	HP-UX	AIX	Tru-64
Personal database server	✔	✔	✔	✔	✔
Network database server	✔	✔	✔	✔	✔
ODBC clients [11]	✔	✔	✔	✔	✔
Embedded SQL clients	✔	✔	✔	✔	✔
SQL preprocessor	✔	✔	✔	✔	✔
Open Client clients	✔	✔	✔	✔	✔
JDBC clients (jConnect driver)	✔	✔	✔	✔	✔
Java in the database	✔	✔	✔	✔	✔
Remote data access [12]	✔	✔			
External stored procedures	✔	✔	✔	✔	✔
iAnywhere JDBC driver [13]	✔	✔	✔	✔	✔

Linux Note for Adaptive Server Anywhere During testing, crashes have been observed under the 2.4.2 SMP kernel. Later 2.4.x kernels do not exhibit this problem.

Under some conditions, the EAX register may be overwritten with an incorrect value. The EAX register is used to hold a repeat count for memory copy operations, and so the bad value can cause large amounts of memory to be overwritten, including memory holding data structures being used by other threads.

Testing with the 2.4.7 and 2.4.9 SMP kernels reveals that the problem has been fixed. While the 2.4.2 kernel can be used with a single processor, please upgrade to the 2.4.7 or 2.4.9 kernel for SMP support.

[11] ODBC driver manager may not be available. May have to link directly to ODBC driver.

[12] ODBC remote data sources requires iAnywhere Solutions ODBC driver. For a list of supported platforms, see "iAnywhere Solutions ODBC drivers supported platforms" on page 135.JDBC remote data sources require a JDBC driver.

[13] Requires JRE 1.4 or higher on Itanium chips. Requires JRE 1.3 or higher on other chips.

UltraLite host platforms

Component	Solaris	Linux	HP-UX	AIX	Tru-64
SQL preprocessor (Embedded SQL only)	✔	✔			
UltraLite generator (C++ API, Pure Java)					

MobiLink synchronization server

Component	Solaris	Linux	HP-UX	AIX (4.3.3 or later)	Tru-64
MobiLink synchronization server	✔	✔		✔	
TCP/IP synchronization	✔	✔		✔	
HTTPS synchronization	✔	✔		✔	
HTTP synchronization	✔	✔		✔	
Transport-Layer Security	✔	✔		✔	
Java synchronization logic	✔	✔		✔	
.NET synchronization logic					

Adaptive Server Anywhere MobiLink client

Component	Solaris	Linux	HP-UX	AIX	Tru-64
ASA MobiLink clients (*dbmlsync*)	✔	✔			
TCP/IP synchronization (client)	✔	✔			
HTTPS synchronization (client)	✔	✔			
HTTP synchronization (client)	✔	✔			
MobiLink extraction utility (*mlxtract*)	✔	✔			

SQL Remote

SQL Remote components that operate against Adaptive Server Enterprise databases require Sybase Open Client and/or Open Server libraries. These libraries are not available on Compaq Tru-64.

Component	Solaris	Linux	HP-UX	AIX	Tru-64
ASA Message Agent (*dbremote*)	✔	✔	✔	✔	✔
ASE Message Agent (*ssremote*)	✔	✔	✔	✔	
ASE stable queue (*ssqueue*)	✔	✔	✔	✔	
ASA extraction utility (*dbxtract*)	✔	✔	✔	✔	✔
ASE extraction utility (*ssxtract*)	✔	✔	✔	✔	
File message type	✔	✔	✔	✔	
FTP message type	✔	✔	✔	✔	✔
MAPI message type					
SMTP message type	✔	✔	✔	✔	✔

Replication agent

The Replication Agent requires Sybase Open Client and Open Server libraries.

Component	Solaris	Linux	HP-UX	AIX	Tru-64
Replication Agent	✔	✔	✔	✔	

Administration tools

Component	Solaris	Linux	HP-UX	AIX	Tru-64
Command line administration utilities	✔	✔	✔	✔	✔
Sybase Central	✔	✔			
Interactive SQL	✔	✔			

With the exception of the command-line administration utilities, the administration tools employ a Java 2 runtime environment, version 1.3.1.

The following paragraphs are taken from the Java 2 SDK, Standard Edition Version 1.3.1 system requirement for Solaris, and apply to the administration tools:

The Java 2 SDK, Standard Edition, v. 1.3.1 (J2SDK 1.3.1) is intended for use on Solaris 2.6, Solaris 7, Solaris 8, and Solaris 9 Operating Environments.

Prior to installing the Java 2 SDK, you should insure that you have installed the full set of required patches needed for support of this release. To obtain patches, see the SunSolve support website. You will find a patch cluster for each Solaris operating environment platform. Each patch cluster applies to all supported versions of the Java 2 Standard Edition (J2SE) on the given platform.

See also Solaris Font Package Requirements for information about which font packages should be on your system.

MobiLink synchronization consolidated databases

MobiLink supports the following consolidated databases:

Database server	Version
Adaptive Server Anywhere	Current
Sybase Adaptive Server Enterprise	11.5 and later
Oracle	8i and 9i
Microsoft SQL Server	7 and 2000
IBM DB2	7.1 UDB

Dependencies

For the MobiLink synchronization server to connect to a consolidated database, an ODBC driver is required. iAnywhere Solutions provides a selection of ODBC drivers described in "iAnywhere Solutions ODBC drivers supported platforms" on page 135. Some database vendors also provide their own ODBC drivers.

For a list of ODBC drivers and their status see *Recommended ODBC Drivers for MobiLink* at *http://www.ianywhere.com/developer/technotes/odbc_mobilink.html.*

iAnywhere Solutions ODBC drivers supported platforms

iAnywhere Solutions ODBC drivers provide ODBC access to third-party databases for the MobiLink synchronization server and for Adaptive Server Anywhere remote data access.

The Adaptive Server Anywhere ODBC driver is not listed in this section. For information on the Adaptive Server Anywhere ODBC driver, see "Adaptive Server Anywhere supported operating systems" on page 125.

iAnywhere Solutions ODBC drivers are available for the following operating system and database combinations:

Operating system	Database
Solaris 7.0, 8.0, and 9.0	Sybase Adaptive Server Enterprise 11.5 and later Oracle 8i and 9i
Solaris 2.6	Sybase Adaptive Server Enterprise 11.5 and later
IBM AIX	Sybase Adaptive Server Enterprise 11.5 and later Oracle 8i and 9i IBM DB2 UDB 7.1
Linux [14]	Sybase Adaptive Server Enterprise 11.5 and later Oracle 8i and 9i

[14] Red Hat, SuSE, and Caldera distributions only.

UltraLite target platforms

The following table lists the supported operating systems for UltraLite deployment, including synchronization streams.

☞ For a list of supported development platforms, see "UltraLite host platforms" on page 126.

☞ For an explanation of the column headings, see "Operating system versions" on page 138.

Component	Windows CE	Palm	Windows XP
UltraLite static C/C++ API	✔	✔	✔
UltraLite Embedded SQL	✔	✔	✔
UltraLite static Java API[15]	✔		✔
UltraLite for AppForge MobileVB	✔	✔	✔
UltraLite ActiveX	✔		✔
UltraLite .NET	✔ [16]		✔ [17]
Native UltraLite for Java	✔[18]		✔
UltraLite C++ component	✔	✔	✔
TCP/IP synchronization	✔	✔	✔
HTTP synchronization	✔	✔	✔
HTTPS synchronization	✔	✔	✔
HotSync synchronization		✔	
ActiveSync synchronization (3.1 and 3.5)	✔ [19]		

[15] Requires 32-bit JDK 1.1.8 or later

[16] Requires .NET Compact Framework version 1.0.3705 or later.

[17] Requires .NET Compact Framework version 1.0.5000 or later.

[18] Requires Jeode VM

[19] Not supported by UltraLite components except Native UltraLite for Java and UltraLite.NET

Component	Windows CE	Palm	Windows XP
Transport-layer security over HTTP or TCP/IP [20]	✔[21] .	✔[22]	✔

Notes

UltraLite dynamic memory requirements mean that devices with Palm OS 3.0, or devices with less than 4 MB of memory, may not run for all but very small database schemas.

[20] Not supported by UltraLite components. Static Java API requires JDK 1.2.2 or greater. JDK 1.4 is not supported.

[21] Pocket PC required

[22] Version 3.5 or later.

Operating system versions

Unless otherwise specified, operating system names correspond to the following versions:

Operating system	**9.0.0**
Compaq Tru-64	4.0D or 5.1 on Alpha processors.
Linux	Linux distributions are all based on defined versions of the kernel, the C library (glibc) and the ncurses library. The supported versions are as follows: ♦ **kernel** 2.2.12 on SMP and UP, 2.4.18 on SMP, or 2.4.2 on UP ♦ **glibc** 2.1.2, 2.1.3, or 2.2 ♦ **ncurses** 4.2 The following Linux distributions are among those that meet the above requirements. ♦ **Red Hat** 6.1, 6.2, 7.0, 7.1, 7.2, 7.3, or Advanced Server 2.1. ♦ **SuSE** 6.3, 6.4, 7.0, 7.2, 7.3, or 8.0 (but not 7.1) ♦ **Caldera** 2.4 or 3.e ♦ **TurboLinux** 6.1, 6.5 or 7.0 ♦ **Mandrake** 7.2, 8.0, or 8.1 Linux distributions are supported on x86-architecture processors. The 64-bit deployment edition is available for Red Hat Advanced Server on Itanium 2 processors.
Hewlett Packard HP-UX	11.0 or 11i, 32-bit version on PA-RISC processors. HP-UX 11i is also supported in a 64-bit version on Itanium 2 processors.
IBM AIX	4.3.1 [23] , 4.3.2, 4.3.3, 5.1, or 5.2 on Power PC processors
Microsoft Windows 98/Me	All versions supported on x86-architecture processors

[23] Patch 1X81739 recommended

Operating system	9.0.0
Microsoft Windows NT/2000/XP	The following are supported: ♦ Windows NT 4.0 or later ♦ Windows 2000 ♦ Windows XP, including Windows XP embedded and Windows XP Tablet PC on x86-architecture processors. A list of components required on the Windows XP Embedded image can be found at *http://www.sybase.com/detail?id=1019835.* ♦ Windows Server 2003, 32-bit versions.
Microsoft Windows (64-bit)	Windows Server 2003, 64-bit versions.
Microsoft Windows CE	Windows CE 3.0 or later operating system on any of the following processors: ♦ MIPS processor ♦ ARM processors, including the XScale series. ♦ x86 processor. UltraLite supports the x86 processor for the Pocket PC 2002 emulator; Adaptive Server Anywhere supports x86 devices and emulation. The Windows CE emulator is also supported for development purposes. Windows CE 3.0 or later includes support for Pocket PC, including Pocket PC 2002, as well as Handheld PC.
Novell NetWare	5.1 with service pack 6 and 6.0 with service pack 3 on x86-architecture processors.
Palm OS	Palm Computing Platform devices running the Palm OS version 3.*x* or later.
Sun Solaris	7.0 [24] , 8.0, or 9.0 32-bit or 64-bit operating systems on SPARC processors.

[24] Patches 106541-14 and 106980-13 recommended to resolve observed SIGIO problems.

Index