SYBASE®

TECHNICAL WHITE PAPER

Synchronizing Data Among Heterogeneous Databases

Principal Author
Robert H. Wiebener, Jr.
Robert.Wiebener@sybase.com

TABLE OF CONTENTS

INTRODUCTION TO HETEROGENEOUS REPLICATION

Not so long ago, a company's data was all stored in a central data repository. Users who needed access would have to create a remote connection or in many cases, required an actual printout of the data. This wasn't too much of a problem when a company acted locally, or had a centralized main office with mainframe computers that controlled the company's data. From this centralized data storage paradigm, the industry has moved to a much more distributed architecture with Client/Server applications becoming the norm. Mergers, acquisitions, and globalization have also necessitated a more distributed data storage model. As companies are acquired or global offices are opened, data needs are often determined at a departmental level. Globalization has resulted in the distribution of data across many different systems, applications, and vendors. This has led to a great many problems, with inconsistent, out-of-date, or duplicate data that can't be trusted or relied upon. Acquired companies or remote offices often make their own decisions about, or bring with them, databases from many different vendors. This further complicates the process of maintaining current, relevant, and correct data at each location.

How can a company make sense of this situation, keeping their data current, relevant, and correct across many different heterogeneous applications and databases? The databases at remote sites must be kept current with changes between sites or between main and standby databases. Sybase Replication Server® has all of the features you need to keep your databases in synch.

This paper will describe Sybase Replication Server and how it enables the user to keep heterogeneous databases synchronized.

REPLICATION SERVER AND DATA REPLICATION

Introduction

What is data replication and how has it changed over the years? Once a simple technology, data replication has evolved into something quite sophisticated as technologies and data needs in the enterprise have grown.

Early Data Replication: Dump and Reload

The concept of replicating data across an organization is not new. Organizations have been distributing copies of their data by "dumping" (downloading, perhaps to tape) and then "reloading" the data on a different site for years. However, the data at each site is not timely and the process is often manual, not automated.

Synchronous Replication: Two-Phase Commit

The introduction of two-phase commit technology in the late 1980s, which Sybase pioneered, provided a more timely way of distributing and sharing corporate data. Two-phase commit allows the synchronization of distributed data, but at a price. Individual component outages can quickly disable the two-phase commit solution, leading to outdated data.

Non-Transaction-Based Replication: Snapshots

Table snapshot implementations allow the asynchronous distribution of changes to individual tables, subsets of tables, or even collections of tables according to a pre-determined schedule. While more advanced than "dump and reload" procedures, the use of table snapshots has several significant drawbacks. Snapshots:

- Do not maintain the integrity of transactions, only provide 'read-only' copies that cannot be updated by the business units that access them,
- Copy individual data tables or data items without maintaining the atomicity of a transaction,
- Only provides a one-way path for data,
- The copies are read-only and no changes can be made to the distributed data.

Non-Transaction-Based Replication: Database Triggers

In addition to snapshots, triggers offer another asynchronous mechanism provided by some database vendors for the purpose of one-way data replication. In early trigger-based replication systems, customers were expected to assemble their own replication applications using these triggers. While offering customers more flexibility than snapshots, early trigger-based replication systems did not overcome the fundamental flaws of snapshot technology.

Transaction-Based Replication with Triggers or Rules

To solve the transactional problems inherent in triggers, processes were introduced that grouped data changes into transactions after a trigger event occurred inside the source database or rule-based replication system.

However, triggers impose a performance and administrative overhead to the source of the data. Such overhead is necessary when triggers are used to protect the integrity of data or to enforce business rules. Early on, Sybase recognized the limitations of trigger-based replication and chose to implement a flexible replication architecture that does not burden database administrators and system performance with triggers.

State-of-the-Art Replication: Sybase Replication Server

After seeing some of the early replication alternatives listed above, organizations quickly discovered that a reliable replication system must do much more than simply copy a piece of data. The system must also:

- Maintain the relational integrity of the data at the transaction level
- Deliver data quickly and efficiently across the network
- Allow distributed sites to modify data and propagate changes
- Be easy to monitor and manage
- Transfer data in any direction across heterogeneous data sources
- Perform automatic data type conversions between heterogeneous data sources

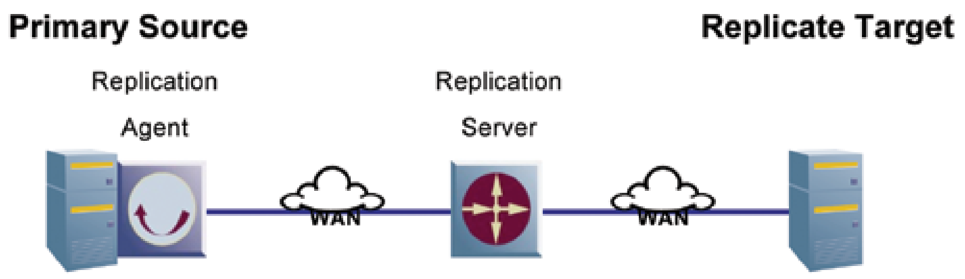
The introduction of Sybase Replication Server in 1989 brought the first and to this date the most sophisticated such product to the market. Working closely with its customers, Sybase created a new, practical, and reliable alternative for the distribution of corporate information. Sybase Replication Server introduced several important elements to the concept of data replication. The asynchronous operations of the product can be divided into six components:

1. Transactions are detected and captured by the replication system in one or more source databases where changes are being made with minimal performance overhead. No triggers or rules are used for this process.
2. The information is distributed across local or wide area networks in a transactional and reliable manner that allows administrators to make the best use of their network resources.
3. Data is translated according to built-in and user-defined data type conversions rules. Built-in rules cover most data translation needs while user-defined data rules allow for flexibility and extendibility.
4. Transactions are delivered to one or more destinations without limiting the flexibility of autonomous users at these destinations to customize or modify the replicated data.
5. A powerful graphical systems management tool allows administrators to monitor and manage every component, or collections of components, of the replication system from a single central site.
6. If a network or system component fails, data in the delivery process is stored temporarily in disk queues. When the component that failed returns to operation, replication restarts and the queued data resume their delivery path. Therefore, no data loss will occur.
7. Using Sybase products, customers can build applications that replicate data changes in any direction across different hardware platforms and heterogeneous data sources.

Architecture

To understand heterogeneous replication, we must first understand the components that make up Sybase Replication Server. In its simplest form, there are two sides to replication, the source and the target. The source system is the primary data server, and is the master data source. When a change occurs in the source, it is recognized by the replication components and transmitted to the target, the replicate data server, where the same change is applied. The components that are responsible for recognizing the change in the source, transmitting it, and applying the change to the target are the replication agent and replication server.

The diagram below shows the relationship between the source and target, as well as the agent and server that will manage the changes between the two systems.



The agent is responsible for recognizing that a change has occurred in the source system. It then transmits the change to the server. The server then applies this change to the target system, keeping the two databases synchronized.

This is, of course, a very simplified explanation as there are many possible configuration options. More commonly, you will have multiple sources, agents, replication servers, and targets. At its core, however, all replication scenarios boil down to variations of this configuration.

Achieving Transactionality Easily

As we know, when data changes, you can't just look at a specific table. Data changes routinely span multiple tables. If you are watching tables with triggers or other methods, you may not be aware of all of the changes applied by a single SQL statement. The ultimate source for monitoring changes is a database's transaction log. Replication Agent uses the transaction log(s) in the primary database to determine what has changed. This greatly simplifies the configuration that must be done by the end-user. Sybase Replication Server's approach differs from triggers because, as was described before, the user would has to do all of the configuration and administration of the triggers and transactions. This can lead to many problems and errors such as transactional integrity issues. With the agent reading the transaction log directly, the configuration is simplified and mostly automatic.

HETEROGENEOUS DATA REPLICATION

What's Different?

Homogenous replication, or same vendor replication is defined as replication between two or more of the same types of databases. This is a much simpler replication environment. You don't have to worry about SQL, data type, communication, or other differences. The term heterogeneous replication refers to replicating data-changing operations between two databases of different vendors. Since the databases are different, we have to resolve the technical differences between data servers. Sybase Replication Server supports the following servers:

- Adaptive Server® Enterprise
- DB2
- Oracle
- Microsoft SQL Server
- UDB

In heterogeneous replication, you can use one or more of these data servers as a source and one or more of these data servers as a target. With Sybase Replication Server, it's easy to mix and match data servers as circumstances and needs require.

Two additional components provided are required to implement a replication system with non-ASE data servers:

- A separate replication agent for each non-ASE source database
- Enterprise Connect Data Access (ECDA) or a data server whose connectivity requirements are compatible with Sybase Replication Server for each non-ASE replicate database

NOTE: Adaptive Server Enterprise is enhanced to support Sybase Replication Server. All of the data server elements required to support Sybase Replication Server (that is, a data-change capture mechanism in the primary database and system tables and stored procedures in the replicate database) are either built into Adaptive Server Enterprise or enabled by utilities that are provided with the Sybase Replication Server or Adaptive Server software.

How's it Used?

"True" Heterogeneous Replication

"True" heterogeneous replication is a type of "application integration". For a number of reasons, companies find themselves needing to keep two heterogeneous databases synchronized. Mergers, acquisitions and departmental data vendor decisions, all dictate the need for true heterogeneous replication. True heterogeneous replication can solve this problem, along with its added complexities of data-type translation, field mapping, default/literal value substitutions, etc.

Real Time Loading

Often times different phases of processing are stored in different database platforms. For instance, let's say the corporate standard for enterprise data-warehouses (EDW) is Oracle, but Sybase ASE is used for online transaction processing (OLTP) data. It is necessary to keep the two databases in synch in near-real time, not just have the data warehouse rebuilt on a nightly basis. Replication can solve this issue by replicating the changes from the ASE OLTP database into the Oracle EDW.

Packaged Applications

Packaged applications typically only support a limited number of DBMS platforms (possibly not the one you have adopted as a standard) and have schemas that are not well documented to the end-user customer. Upgrades to the packaged application make this worse as few vendors supply a list of schema changes or other modifications to metadata to the end-user customer to support modifying a direct replication scheme.

However, most packaged applications support a messaging API such as iDoc, HL7, etc. Sybase's heterogeneous replication can support integration with these applications through the real-time data services (RTDS) facility which translates data modifications into messages in XML or other formats and supports all of the usual message buses as well as propriety protocols.

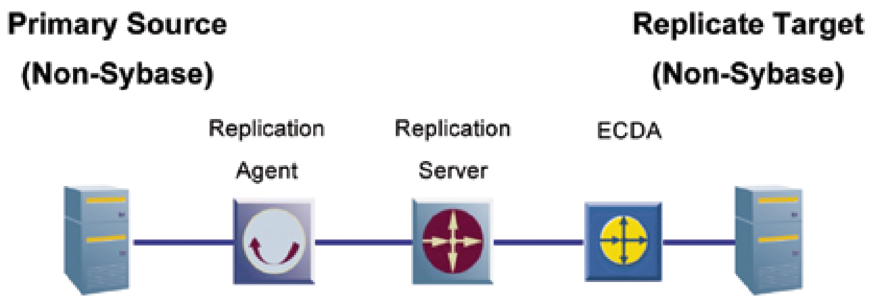
Typically, such integration adds additional workload on the source system. This is caused by the constant inquires made by the message polling mechanism checking for modifications, schema changes, etc. Replication server can exploit the existing disaster recover or reporting stream, and send a copy of the changes to the message bus. This results in NO additional load on the production system.

Internet Integration

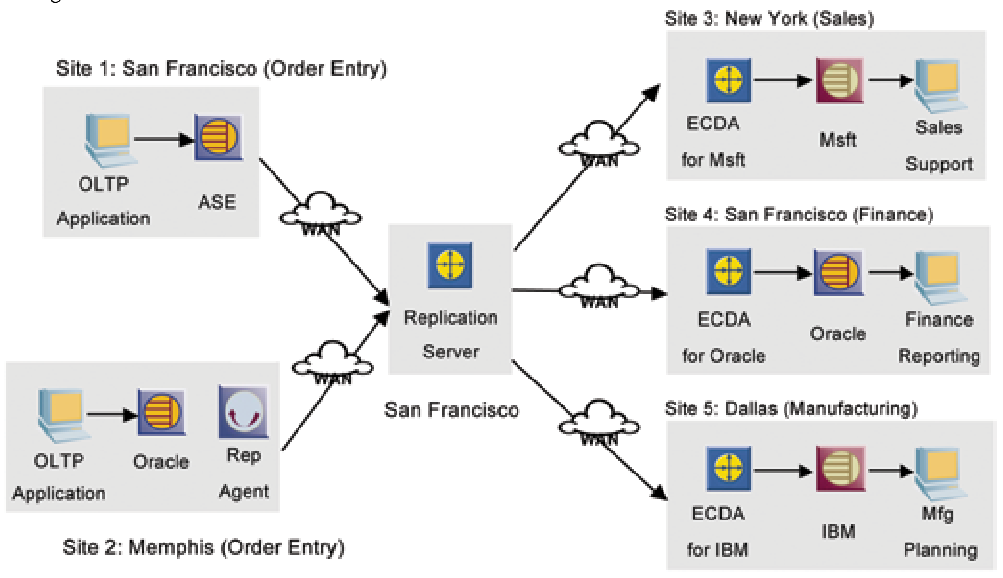
Many internet developers utilize .NET to create their internet applications. Because of the tight integration between .NET and MS SQLServer, developers often find it easier to use MS SQLServer as the underlying database. However, the internal applications that the web users want data from are most often running on enterprise databases such as Sybase, Oracle or DB2. Porting the internal applications to use MS SQLServer is an expensive and sometimes technically daunting task. Utilizing heterogeneous replication, the developers can create replication between the MS SQLServer and the enterprise databases. Updates to addresses, accounts and personal information would be replicated from the web application database (MS SQLServer) to the enterprise databases (Sybase, Oracle, DB2). Further, personal information, account numbers, etc. could be obfuscated so that hackers could not obtain "real" information from the less-secure web database exposed outside the firewall.

Architecture

The architecture diagram for a heterogeneous replication system is very similar to the homogenous replication system diagram shown before. As you can see in the diagram below, the only addition is the ECDA system. The only change is that the replication agent is now no longer a part of the Primary Source, instead it is a separate process. These are the only two physical changes necessary to configure a heterogeneous replication system. If either the Primary Source or the Replicate Target is a Sybase ASE data source or target, then the separate replication agent or ECDA process would not be required as the Sybase ASE data source or target has the necessary components built in.



While this is a very simple configuration, by adding sources and targets, along with the necessary agents, servers, and ECDA processes, you can build a complicated replication system. For instance, you could end up with a configuration such as this:



In this scenario, the Order Entry departments in San Francisco and Memphis add new orders to their ASE and Oracle databases. The built-in ASE replication agent in San Francisco and the stand-alone Oracle replication agent in Memphis would recognize the new orders and send the transactions to the San Francisco replication server. This replication server would then send the transaction to New York Sales, San Francisco Finance, and Dallas Manufacturing data servers. Each of these systems is running on non-Sybase databases. Each of the remote ECDA processes would then apply the transaction to their respective databases, with the data elements needed, thus keeping their databases current and correct.

Replication Agent

A **replication agent** transfers transaction information, which represents changes made to data schemas and execution of stored procedures, from a primary data server to a replication server, for distribution to other (replicate) databases.

In Adaptive Server Enterprise, an embedded replication agent is provided with the database management system software. The replication agent for ASE is called replication agent, and it is an Adaptive Server thread.

For non-ASE data servers, Sybase provides the following replication agent products:

- Replication Agent for DB2 UDB – provides primary data server support for IBM DB2 UDB servers that run on IBM z/OS platforms.
- Replication Agent – provides primary data server support for DB2 UDB, Microsoft SQL Server, and Oracle data servers that run on Linux, UNIX, or Microsoft Windows platforms.

A replication agent is required for each database that contains primary data or for each database where replicated stored procedures are executed.

Replication agent consists of a set of components that work together to perform all the operations required to propagate transactions from a primary database for replication.

The following are the main replication agent components:

- Log Reader – reads the transaction log in the primary database to retrieve transactions for replication.
- Log Transfer Interface (LTI) – generates Log Transfer Language (LTL) and sends it to the primary replication server.
- Log Administrator – administers the replication agent transaction log and manages transaction log objects.
- Log Transfer Manager (LTM) – manages all the other components and coordinates their operations and interactions.

The process is as follows:

1. The Log Reader component retrieves transaction data from the transaction log in the primary database.
2. The Log Reader generates change set data and passes the change sets to the LTI.
3. The LTI component processes the change set data from the Log Reader and generates the LTL to send to the primary replication server.

ECDA

All Sybase ECDA Options provide basic communication connectivity to non-ASE data services. In particular, they provide access management, copy management, and remote systems management. ECDA allows the replication server to apply the changes captured by the replication agent to non-Sybase replicate data servers. ECDA manages the connection to the non-Sybase data server, accepts changes, and applies those changes to the data server as appropriate.

From the replication server's perspective, there is nothing to distinguish an ECDA gateway from an Adaptive Server replicate database. Sybase Replication Server delivers the same commands—and expects the same results—from any ECDA gateway it communicates with.

There are three options available for ECDA:

- ECDA Option for ODBC
- ECDA Option for Oracle
- Mainframe Connect DirectConnect for z/OS Option

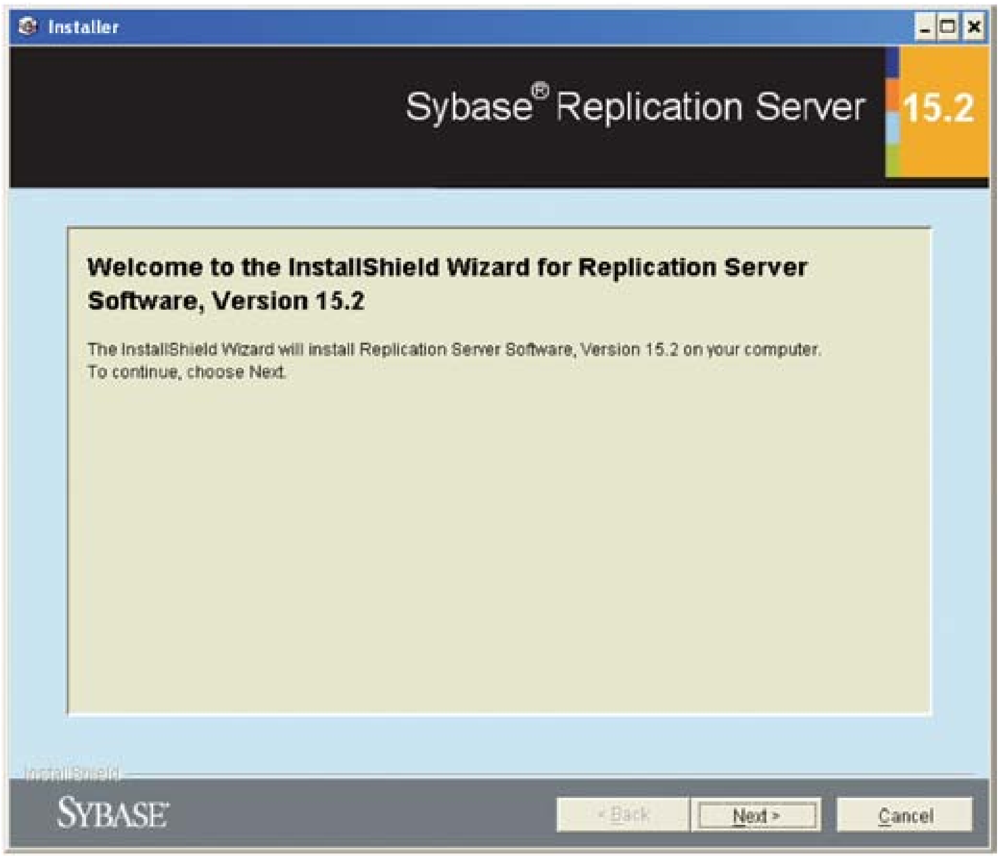
Installation

Installation of a heterogeneous replication system is very straightforward and similar to a Sybase-only replication system with the addition of a few steps. The full process for installing a heterogeneous replication system consists of:

1. Installing the connectivity drivers for the primary database server.
2. Installing the Sybase Replication Agent software.
3. Installing Sybase Replication Server.
4. Configuring the replication server and primary data server connections.
5. Setting up the replication agent instance, defining what data sources will be monitored for changes.
6. Verifying the replication system using replication agent test scripts (optional).
7. Materializing subscriptions to primary data, providing an initial load of data from the source data servers.
8. Starting replication to capture changes and applying them to target systems.

As you can see, except for the addition of the replication agent steps, the steps required are the same as if you were configuring a Sybase-only replication system.

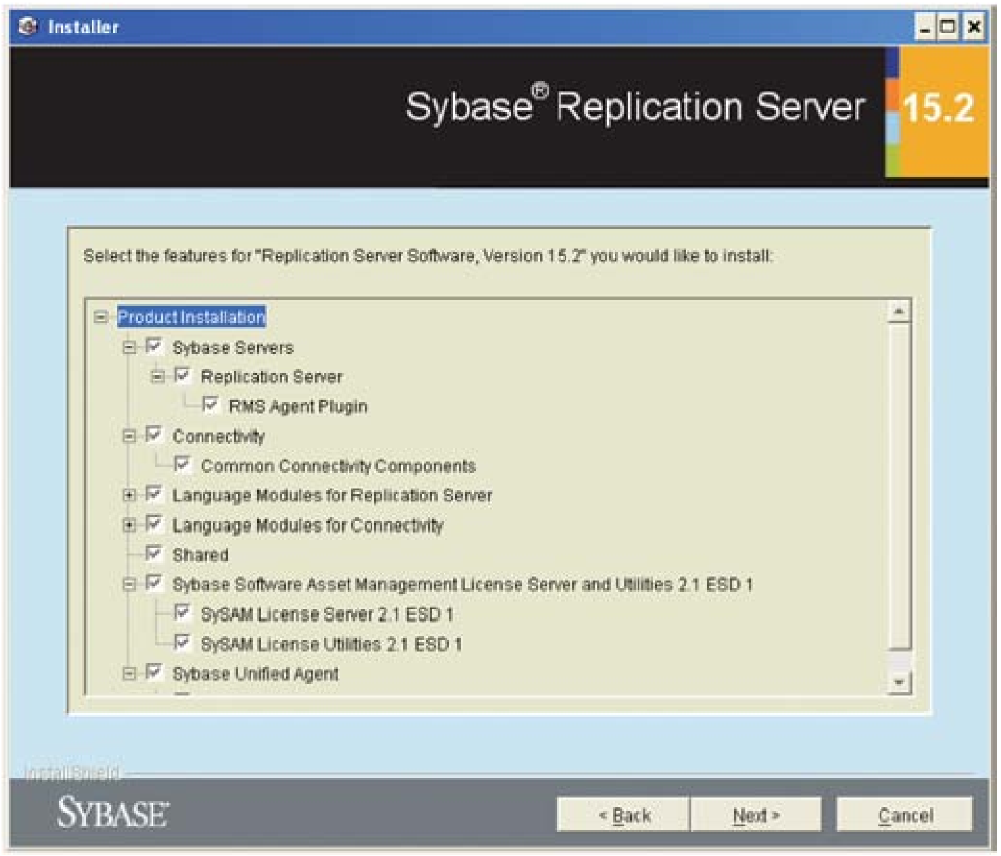
Installation and initial configuration is done via either GUI or command line. InstallShield is used when doing a GUI install. Since InstallShield is java based, the installation on both Linux and Windows-based systems will be similar. The InstallShield wizard will step you through the installation process.



Welcome to the InstallShield Wizard for Replication Server Software, Version 15.2

The InstallShield Wizard will install Replication Server Software, Version 15.2 on your computer. To continue, choose Next.

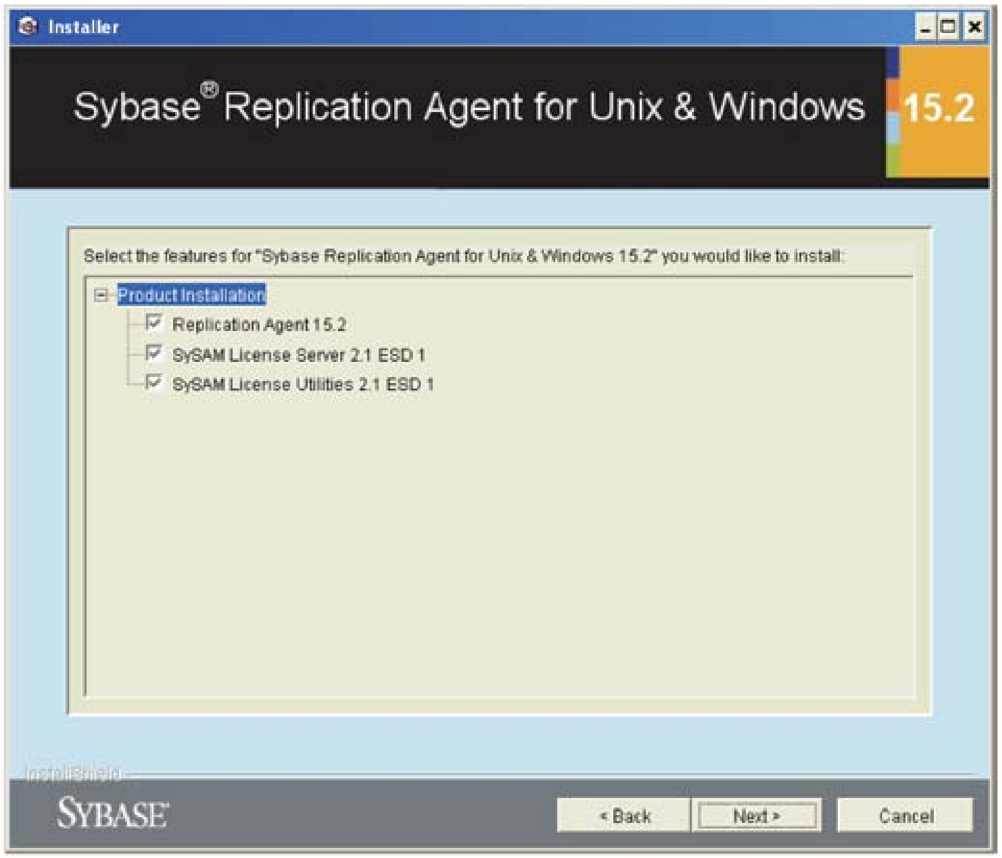
After the usual introductory and license screens, the wizard will allow you to do a Typical (default), Full, or Custom install. If you choose Custom install, you will be given a choice of components to install:



Select the features for "Replication Server Software, Version 15.2" you would like to install:

- Product Installation
  - ☑ Sybase Servers
    - ☑ Replication Server
      - ☑ RMS Agent Plugin
  - ☑ Connectivity
    - ☑ Common Connectivity Components
  - ☑ Language Modules for Replication Server
  - ☑ Language Modules for Connectivity
  - ☑ Shared
  - ☑ Sybase Software Asset Management License Server and Utilities 2.1 ESD 1
    - ☑ SySAM License Server 2.1 ESD 1
    - ☑ SySAM License Utilities 2.1 ESD 1
  - ☑ Sybase Unified Agent

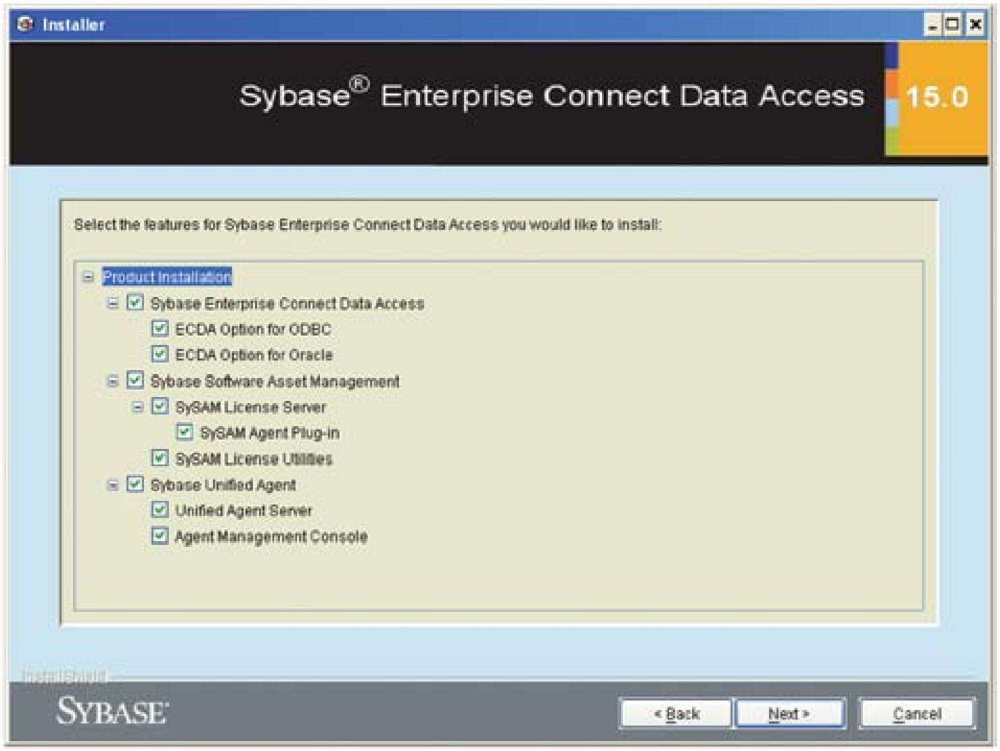
Installing the Replication Agent for the non-Sybase data source is also accomplished by using either a GUI (InstallShield) or command-line install script. The Replication Agent installation GUI is very similar to the Replication Server Installation GUI and allows you to choose which options to install.



Select the features for "Sybase Replication Agent for Unix & Windows 15.2" you would like to install:

- Product Installation
  - ☑ Replication Agent 15.2
  - ☑ SySAM License Server 2.1 ESD 1
  - ☑ SySAM License Utilities 2.1 ESD 1

If you were going to use a non-Sybase data target such as Oracle or Microsoft SQL Server, you would also have to install and configure ECDA to be able to apply changes to a non-Sybase data target. ECDA is also installed using InstallShield GUI or command-line scripts.



Select the features for Sybase Enterprise Connect Data Access you would like to install:

- Product Installation
  - ☑ Sybase Enterprise Connect Data Access
    - ☑ ECDA Option for ODBC
    - ☑ ECDA Option for Oracle
  - ☑ Sybase Software Asset Management
    - ☑ SySAM License Server
      - ☑ SySAM Agent Plug-in
    - ☑ SySAM License Utilities
  - ☑ Sybase Unified Agent
    - ☑ Unified Agent Server
    - ☑ Agent Management Console

PowerDesigner® can be used to assist with the design, configuration, deployment, and documentation of a replication system. You can easily reverse-engineer existing data servers, choose what data to replicate, and decide which targets will receive the updates. PowerDesigner will then generate scripts that will ease the configuration and deployment of the replication system. Finally, PowerDesigner can be used to document the entire replication system, making future changes much easier.

Replication Functionality

- Sybase Replication Server is capable of replicating:
- DML – data modification language
- DDL – data definition language
- DCL – data control language
- Stored Procedures

DML

Sybase Replication Server can replicate all logged data modification commands such as: insert, update, Delete, Slow BCP, etc. As these commands are issued, Replication Agent will recognize that a change has occurred, after commit, and send the change to a replication server. Depending on the configuration, the replication server will either forward the change to all subscribing clients or it will apply the change to the data server. The data will then be applied to the replicate data server, completing the change and maintaining the synchronization of the source and replicate data servers. The replication agent will be a separate process and ECDA will supply the communication and connectivity to non-ASE data servers. However, this process is the same, regardless of what type of data server is the source or the target.

DDL

DDL changes, such as: create, alter, drop objects, DDL stored procedure and even replication server commands such as sp_setreptable and sp_reptostandby, can all be replicated. When replicating DDL between heterogeneous data sources and replicates, no translation or adjustment of DDL commands are provided by the replication agent. Therefore, DDL replication should only occur between homogenous databases. However, this is a very powerful tool and can reduce a database administrator's workload by propagating schema changes. This also minimizes change control issues by insuring that when changes are made to a source data server's schema, the same changes will be applied to target data servers.

DCL

DCL commands such as: grant, revoke, and security stored procedures can be replicated. As you manage users or permissions, these changes can be applied to replicate data sources. With Sybase Replication Server, it's easy to maintain a stand-by database, without having to apply changes for permissions, users, or security manually.

Stored Procedures

Sybase Replication Server will replicate the execution of a stored procedure. Only the execution is replicated, with the supplied parameter values. The actual commands executed are dependent on the stored procedures that are in place on the primary and replicate data sources.

For a full list of DML, DDL, DCL, and stored procedures that can be replicated, please consult the "Sybase Replication Server Administration Guide".

Data Type Translation

In a heterogeneous replication system, when information is replicated from one data server to another, a subset of data types that are unique to that vendor must be translated. User-created function strings can produce these data type translations, but require significant user input and are limited by the capabilities of the replicate data server.

To make data type translations more readily available for different data servers, Sybase Replication Server provides heterogeneous data type support (HDS), an easy-to-apply methodology for translating data types at the replication server. HDS supports selected data type translations between the following data servers:

- Adaptive Server Enterprise
- DB2
- Oracle
- Microsoft SQL Server
- UDB

When you use HDS, you can choose which columns and data types in the primary database are to be translated, and which replicate data servers will receive the translations.

The replication agent for each data server delivers replicate values to the replication server in a data type format that the replication server understands, which includes the literal value, delimiter information and other data type attributes. The replication server handles the value as its base data type—one of the native replication server data types described in the Sybase Replication Server Reference Manual.

You can implement data type translations in two ways:

- **Class-level translations** – translate all instances of a data type for a particular connection.
- **Column-level translations** – translate all instances of a column described by a table replication definition.

You can verify how translations alter values before you set up column- or class-level translations by utilizing system administration functions. These functions accept a value, a source and target data type and return the target value. It is most useful with the diagnostic version of Sybase Replication Server, which, if the translation fails, allows you to trace the reason for the failure.

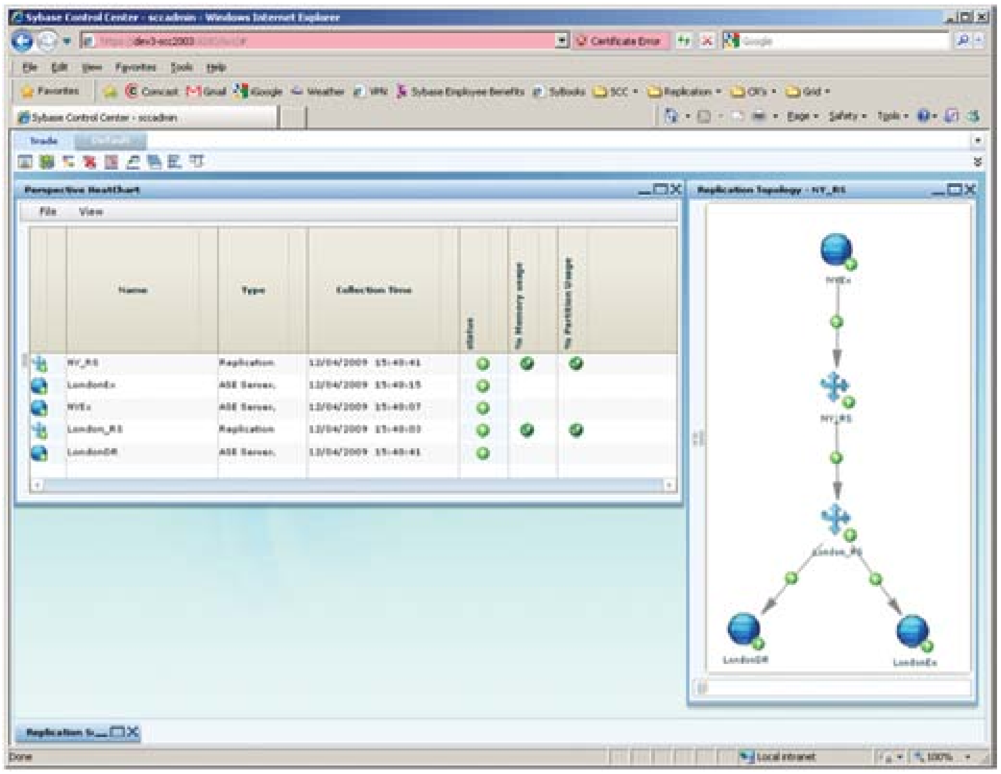
Monitoring Your Replication Environment

After you've built your replication environment, you need to be able to see what's working, what's gone wrong, and what needs to be fixed. Sybase Control Center (SCC) is a Web-based solution that allows you to monitor and control the status and availability of servers in large, complex, and geographically dispersed replication environments. It provides status information at a glance, using server dashboards and a heat chart for displaying the availability or status of a specific server. SCC has alert notification to notify the user when an event occurs in the replication environment. Alerts can be sent through email or on a JMS message bus when the state of a server or component changes, or when a user-defined threshold is reached.

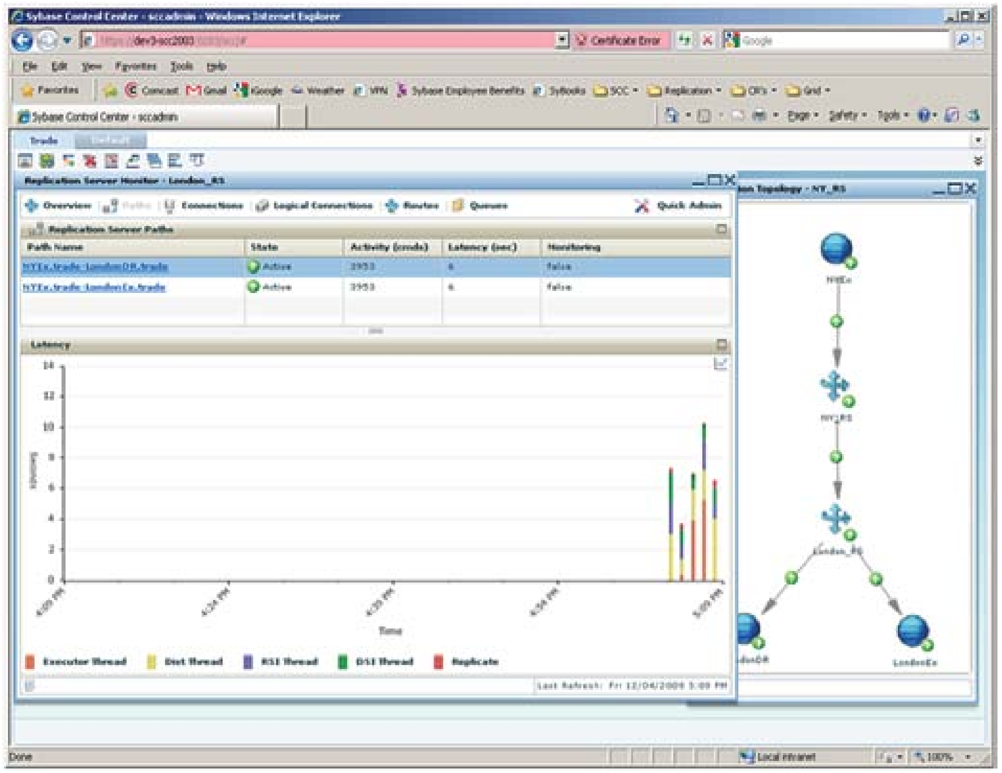
The elements that you can monitor in your Sybase Control Center replication environment are:

- **Data Servers** – data source or targets
- **Replication Server**
- **Replication Agent** – for ASE or non-ASE data servers
- **Components** – objects in a server in a replication environment. Examples of components in a Replication Server are connections, routes, and queues
- **Replication Path** – the set of servers through which transactions pass when moving from the primary to the replicate database

This example shows SCC monitoring a replication environment. A view of the system queues, along with a topology of the environment is displayed. The topology shows that the connection to the data source at the lower right is currently displayed.


Sybase Control Center - sccadmin - Windows Internet Explorer
Certificate Error
File Edit View Favorites Tools Help
Favorites
Sybase Control Center - sccadmin
Trade
Perspective HeatChart
File
View
Replication Topology - NY_RS
NYEx
NY_RS
London_RS
LondonDR
LondonEx
Replication S...
Done
Local intranet
100%

	Name	Type	Collection Time	Status	% Memory usage	% Partition Usage
	NY_RS	Replication	12/04/2009 15:40:41	●	●	●
	LondonEx	ASE Server,	12/04/2009 15:40:15	●		
	NYEx	ASE Server,	12/04/2009 15:40:07	●		
	London_RS	Replication	12/04/2009 15:40:03	●	●	●
	LondonDR	ASE Server,	12/04/2009 15:40:41	●		



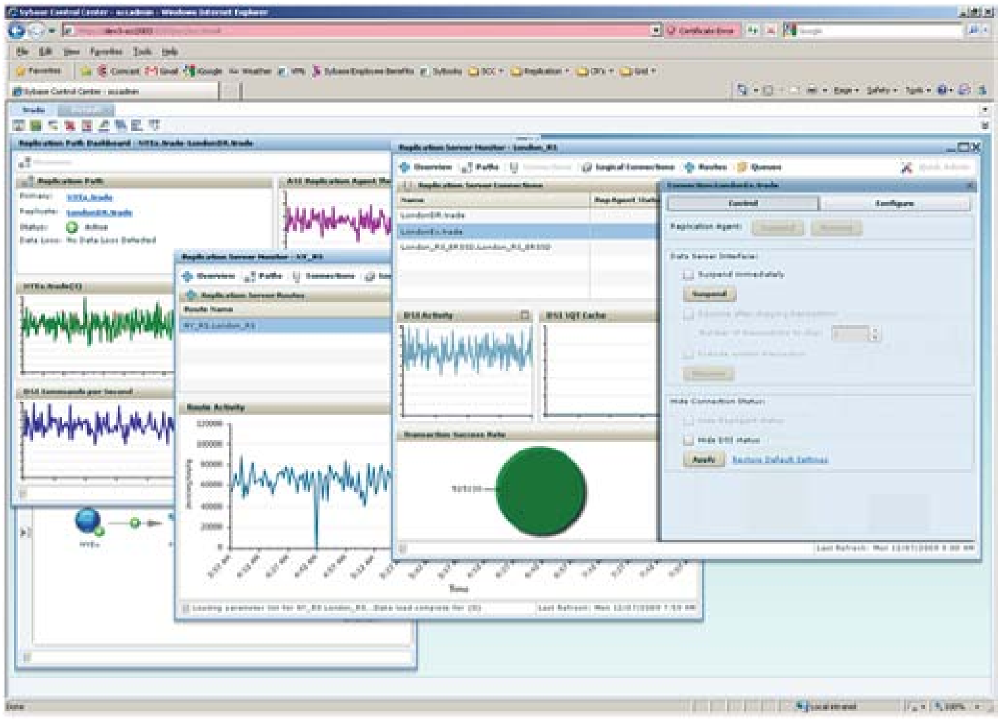
Replication Server Paths

Path Name	State	Activity (cmds)	Latency (sec)	Monitoring
NYEx.trade_LondonDR.trade	Active	3953	6	false
NYEx.trade_LondonEx.trade	Active	3953	6	false

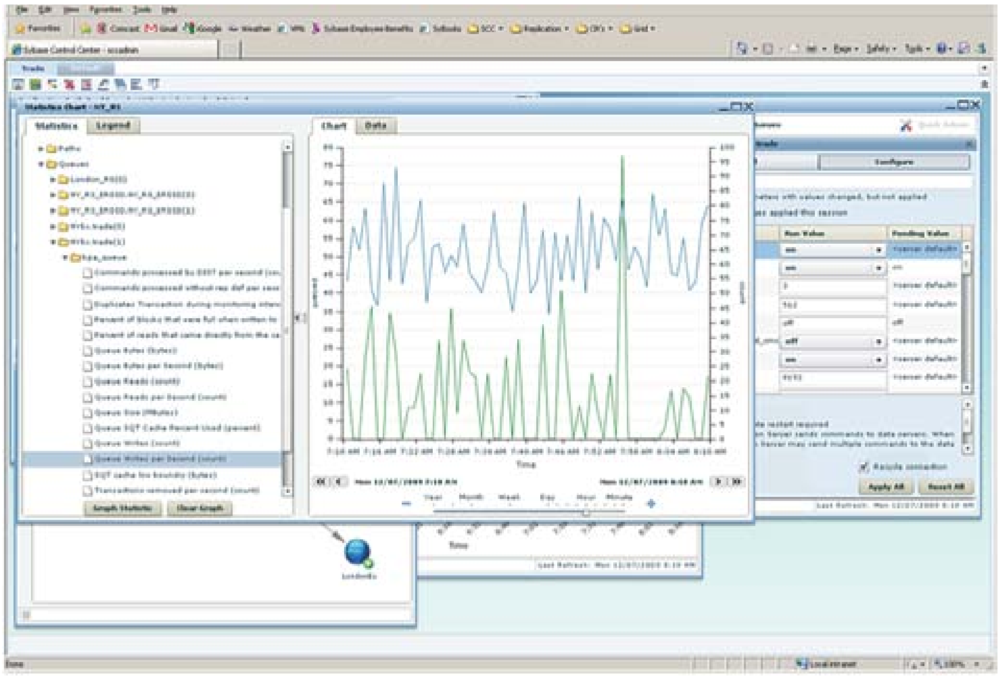
Last Refresh: Fri 12/04/2009 5:09 PM

Using graphical presentation, Sybase Control Center provides a quick view of two important aspects of replication: throughput and latency. You can see the amount of data being transferred (throughput) and the amount of time it takes (latency) all in one dashboard. With such visual help, you can efficiently identify potential problems.

The quick administration screens allow the user to set configuration parameters for a server or component as well as control the flow of data by suspending and resuming components.


Sybase Control Center - sccadmin - Windows Internet Explorer
Replication Path Dashboard
Replication Server Monitor - NY_RS
Replication Server Monitor - London_RS
Route Activity
DSI Activity
Transaction Success Rate
Control
Configure
Suspend
Apply
Restore Default Settings

SCC Historical monitoring allows the user to graph any statistic collected – i.e. Monitors and Counters, rs_ticket, etc. Statistics are displayed in a tree hierarchy and grouped by key performance areas.



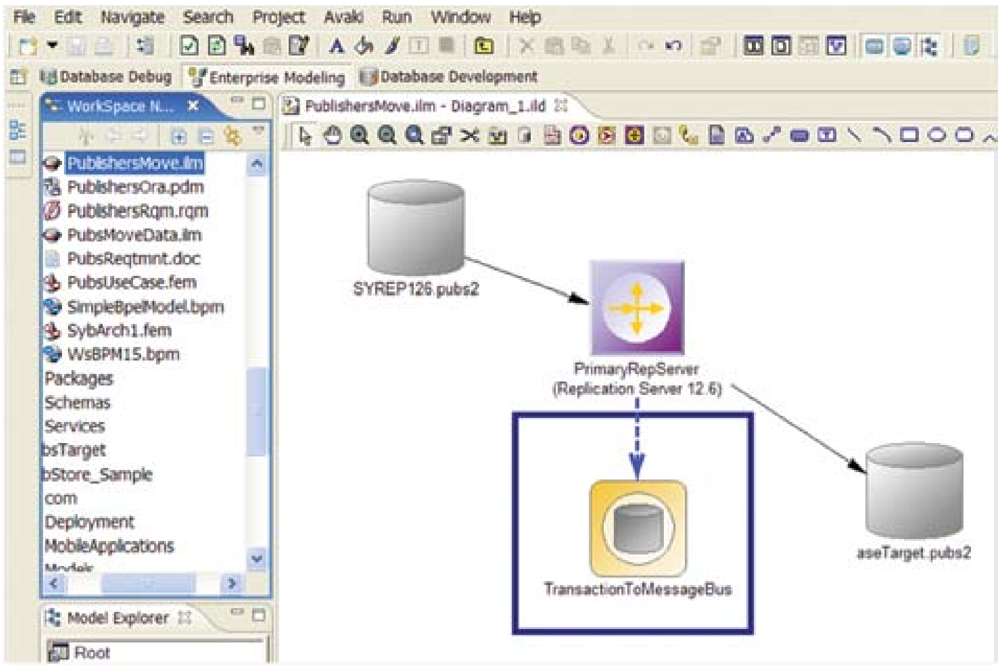
MODELING REPLICATION SYSTEMS TO STREAMLINE DEVELOPMENT

PowerDesigner can be a powerful tool when designing, configuring, deploying or documenting your Replication system. Using PowerDesigner to model your replication system has many benefits including:

- Easily create and configure a replication system using modeling components, reverse-engineering, and/or generated replication server scripts
- Reduce errors due as a result of well-defined model
- Manage changes by having a central system model
- Quickly see the effect of changes on the system
- Visualize and document the replication system

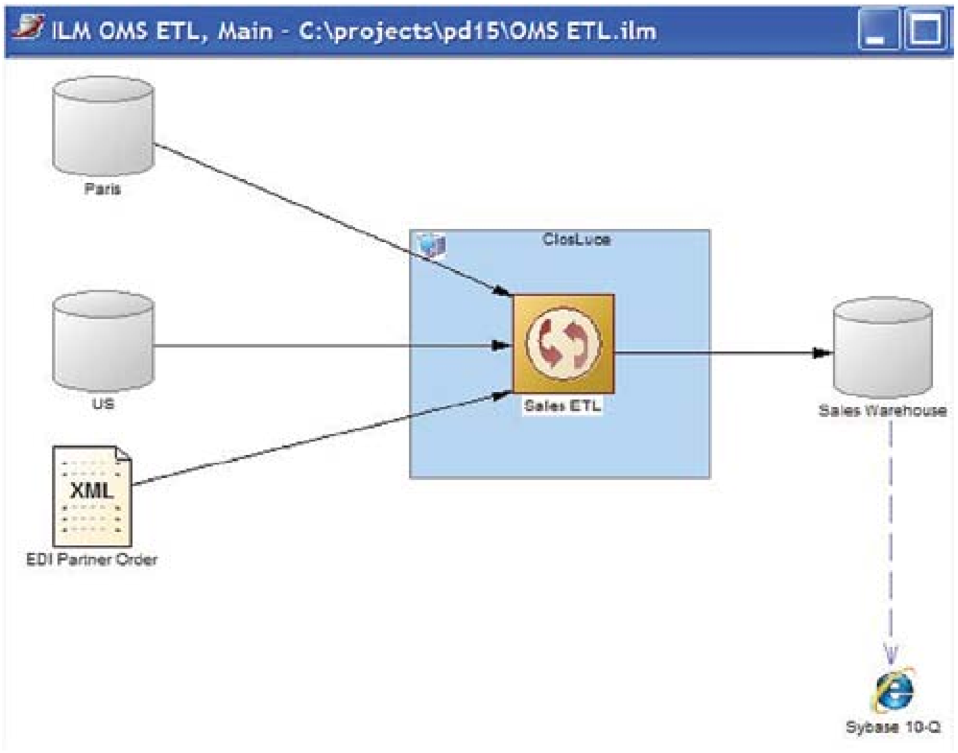
PowerDesigner models all of the components in a replication system, including: network components, data connections, replication definitions and publications and subscriptions. Utilizing the various features of PowerDesigner, you can reverse-engineer the data schemas for your source and target data servers. Then, using these schemas, you can decide which data will be replicated and to which targets. You would then model the replication components: agents, servers, ECDA installations, and configure the connections between them. Finally, you can generate scripts that will allow you to apply the configuration to the replication system components. The PowerDesigner models will document the installation and allow you to understand and modify the system much more efficiently and easily.

As an example, this screenshot shows the PowerDesigner interface, with a simple replication environment modeled:



Replication Integration with ETL

Replication Server can also integrate with Sybase ETL. This allows a more complete view of the entire environment. For instance:



This screen shows multiple data sources with ETL as an intermediate target and a final target of a sales warehouse database.

Heterogeneous Replication Modeling

To model a heterogeneous replication system, PowerDesigner must be configured to model not only the source and replicate databases, but also the replication agent and/or ECDA instances. To model a Heterogeneous Primary Database:

- Add server objects to the PowerDesigner model to describe the primary database and replication process, each with the appropriate host machine name and port number.
- Specify the appropriate properties for the replication server connection and configuration database on the Replication Server Connection tab of the replication process property sheet.
- Specify the replication agent type, the replication agent user (to access the replication server), the primary database user (to access the database), and the other properties on the Replication Agent Options tab of the primary database property sheet.

To model a Heterogeneous Replicate Database:

- Specify server objects to contain the replicate database and replication process, each with the appropriate host machine name and port number.
- Specify the ECDA instance name in the code of the replicate database in its property sheet.

Reverse Engineering an Existing Replication System

PowerDesigner can reverse engineering an existing replication system. To do this, you must reverse engineer the source and replicate databases within PowerDesigner. This is accomplished by using the PowerDesigner **Tools > Reverse Engineering Replication Server** command. The steps to reverse engineer the system are:

1. For each replication process, define the data source, user name and password of the consolidated database in the Database Connection tab of the replication process property sheet.
2. Select Tools > Reverse Engineering Replication Server.
3. Select the replication processes you want to reverse engineer.

For each remote database, PowerDesigner asks you to select the data source of the remote database. A PowerDesigner model will be generated, showing the connections and data used by the system. The system components and designs are no longer on scraps of paper or in someone's head. The model shows exactly how the system is configured and deployed.

Generating Replication Server Scripts from Defined Model

Once you have defined a PowerDesigner model, you can choose to generate SQL scripts for the replication process and/or for the primary and replicate databases. The PowerDesigner **Tools > Replication Server > Generate Scripts** command is used to perform the generation. Each script can be previewed and/or saved as needed, then executed using the isql command.

CONCLUSION

As you can see, heterogeneous replication is not all that different from homogeneous replication. There are many different methods, but Sybase Replication Server is the most powerful, easiest to configure, and has the highest performance of the many replication methods. Sybase Replication Server is a very powerful tool that allows you to quickly and easily replicate data between many different data servers, from many different vendors. Sybase also offers a very powerful modeling tool, Sybase PowerDesigner that assists with development, deployment, and documentation of new or existing replication systems, making the process easier and quicker, both now and in the future.

Sybase, Inc.
Worldwide Headquarters
One Sybase Drive
Dublin, CA 94568-7902
U.S.A
1 800 8 sybase

www.sybase.com

SYBASE®