



Data Access & Data Replication

Data Sheet

Ardent UniVerse RDBMS makes it easy to meet the full range of remote data access and replication needs for today's enterprise database environments. Three separate capabilities are available, each one serving as a perfect match for a specific requirement.

- A distributed database capability offers transparent, updatable access to UniVerse data located in any UniVerse system in a TCP/IP network.
- Data replication provides publication of read-only data from a single system to multiple UniVerse systems.
- Hot standby delivers data integrity and resilience in fail-over situations.

The availability of such specialized solutions makes it possible to implement the right level of control, while minimizing system overhead and maximizing flexibility. UniVerse distributed data access and replication capabilities are available for both UNIX and Windows NT-based UniVerse servers.

Technologies

Today's enterprise database environments can present a full range of distributed data access and replication challenges. While all of these could possibly be met with a comprehensive distributed database approach, the associated complexity, cost, security considerations, etc., would be daunting, especially if the need is only for limited replication or hot-standby capabilities.

The recent addition of the UniVerse Data Replication (UV/DR) services to the UniVerse portfolio of technologies now makes it easy for developers to select the right solution for the right challenge or task.

- Distributed Database

UV/Net allows developers to distribute applications seamlessly across multiple UniVerse servers with complete concurrency control.

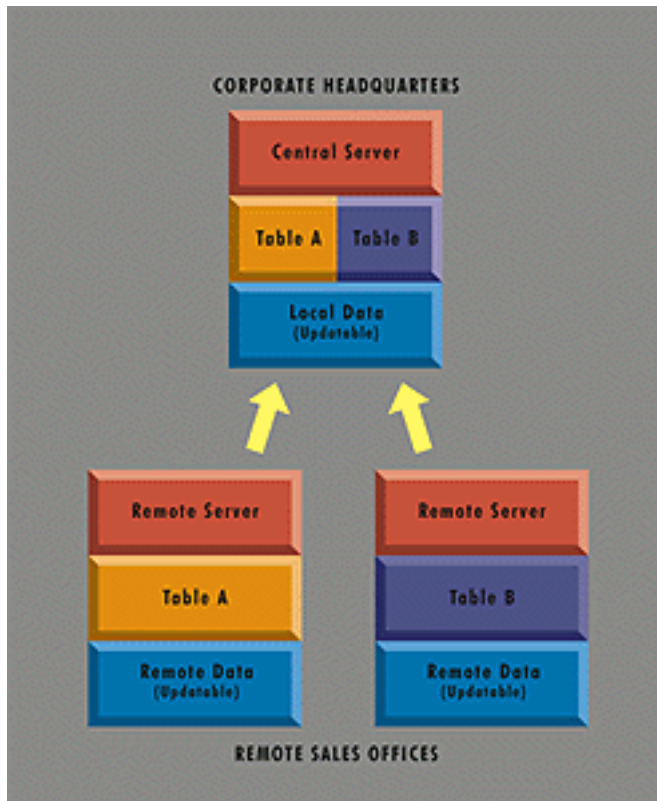
- Database Replication

UV/DR offers a means of copying information from one server to another UniVerse server in a pre-defined manner appropriate to their needs.

- Hot Standby

UV/DR establishes and manages a hot-standby capability, helping reduce application downtime when a hardware failure causes the complete loss of the master application server.

Figure 1: Distributed Database. Access from multiple remote sales offices to corporate headquarters.



The following discussion provides an overview of the features and benefits of each technology, plus specific guidelines to help developers select the right tool for their specific challenges.

Building a Distributed Database

UV/Net integrates separate UniVerse databases into one database that users can access, view, query and update, as if it were local to the data. UV/Net also manages the communication among these distributed databases over local and wide area networks, as well as among diverse hardware systems and disparate releases of the UniVerse RDBMS, to provide a comprehensive approach to distributed data access.

An essential feature of the UV/Net solution is its ability to ensure concurrency control (record locking) between each UniVerse server participating in reads and writes of files at the database level. If a remote server is accessing a local file on another server, then UV/Net manages the lock consistency between systems. Therefore, from a developer's or user's perspective, each file and dictionary can be manipulated through UniVerse BASIC stored procedures and UniVerse Retrieve language. UV/Net also provides an ability to execute remote stored procedures, or commands via RPC requests on remote servers, using BASIC.

Here are two examples of the flexible, distributed database solutions that developers can build with UV/Net:

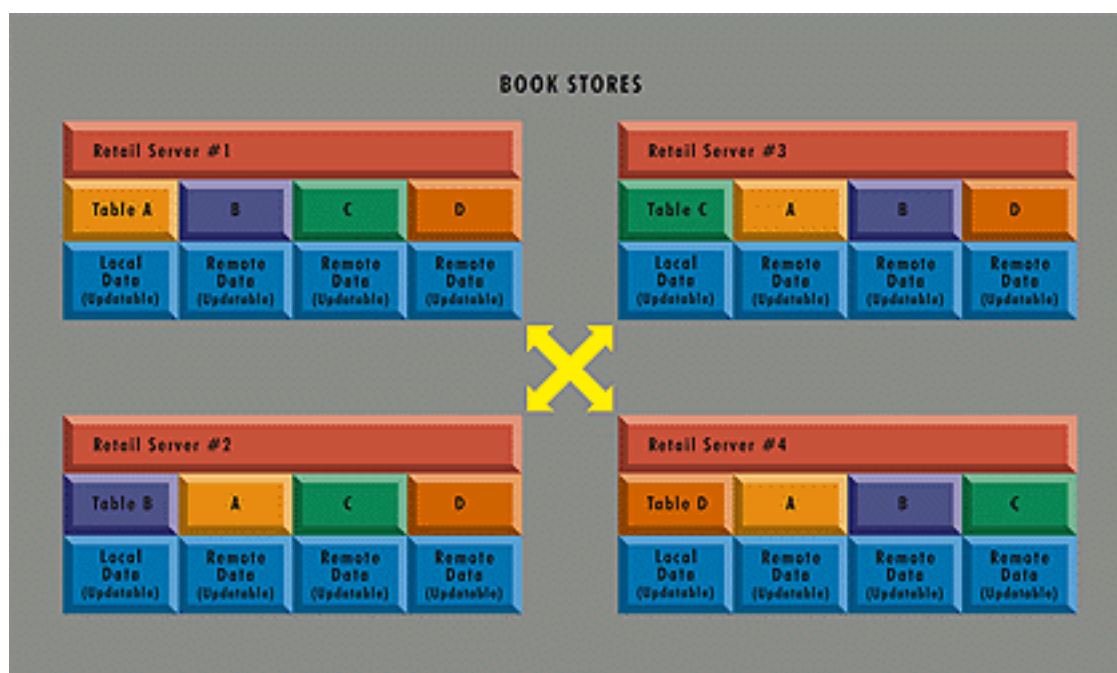
- Access from multiple remote sales offices to corporate headquarters

See Figure 1. An organization may elect to store its entire database at corporate headquarters and provide its remote sites with read/write access to their portion of an entire consolidated customer file. This allows each site to have full access to its segment of the complete corporate data set without creating the security issues that would occur if each site knew or had a copy of the sensitive, complete corporate information.

•• Access to and from multiple stores in a chain

See Figure 2. A chain of four bookstores, each with its own inventory, could use the same inventory tracking application in each of the retail outlets. When a customer requests a book, the staff could check to see if it is in stock locally, or available in one of the other stores. If the publication is located at another store, an internal order could be placed to ship the book to the store closest to the customer.

Figure 2: Distributed Database. Access to and from multiple stores in a chain.



Using Data Replication

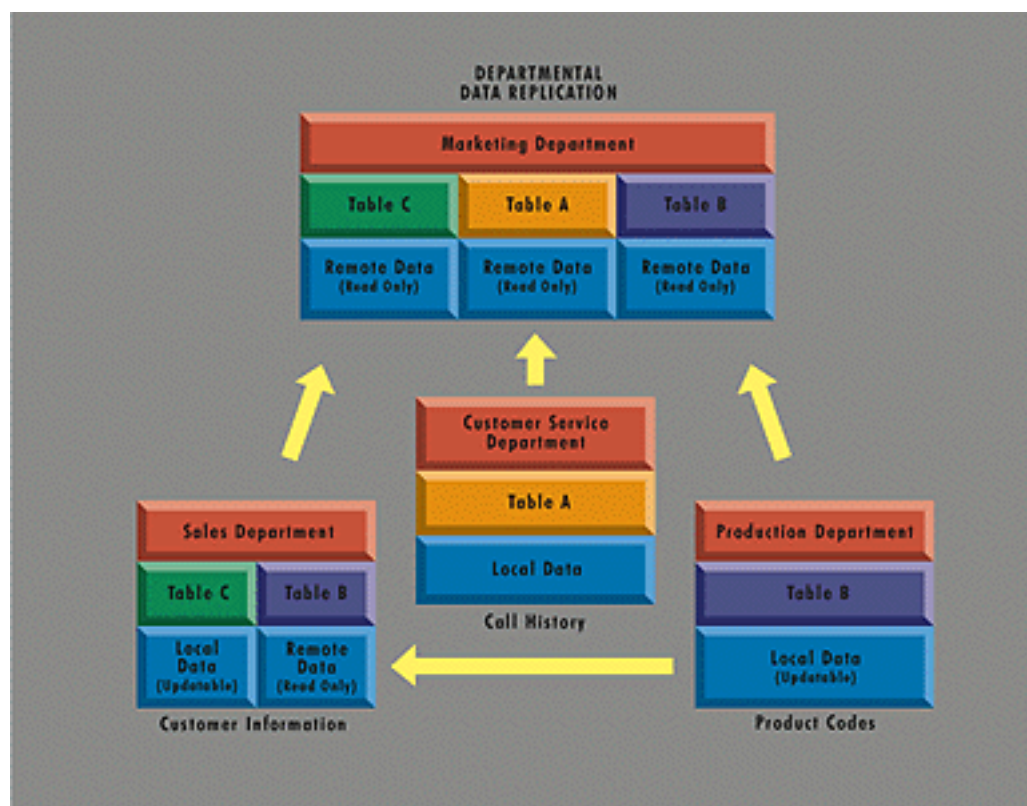
UniVerse Data Replication (UV/DR) provides an automatic and reliable methodology for delivering read-only copies of a source UniVerse database or file to another UniVerse system. The server system offering the source database or file is called the "Publisher." The one or many UniVerse servers requesting copies of the database or file updates from the Publisher are referred to as "Subscribers." A Publisher can also be a Subscriber of another Publisher's database or file updates and vice versa.

In this Publisher/Subscriber architecture, any update made to a source database or file will be propagated to the other systems' target databases or file copies in a pre-determined time by each Subscriber system. In addition, if database transactions are involved, only updates that are part of committed transactions are distributed. The graphical UniVerse Admin (UV/Admin) tool handles the configuration of each Publisher and Subscriber system from a networked PC. It configures each Subscriber's consumption of data on a system-wide basis. The three available options are:

- a continuous update mode
- a batch mode (snap shot) that applies multiple, previously stored updates all at once at a preset time, or
- a batch mode that applies updates when initiated manually.

With UV/DR, developers can quickly and easily build a number of replication solutions. For example, Figure 3 shows a peer-to-peer configuration where departmental servers on a LAN distribute copies of their source data, which does not require simultaneous updates from another location, between interested departments without manual intervention. In this instance, the marketing department is taking copies of customer service's call history, the sales force's customer database, and production's product codes. At the same time, the sales department is using production's product codes.

Figure 3. Peer-to-Peer configuration of departmental servers on a LAN.



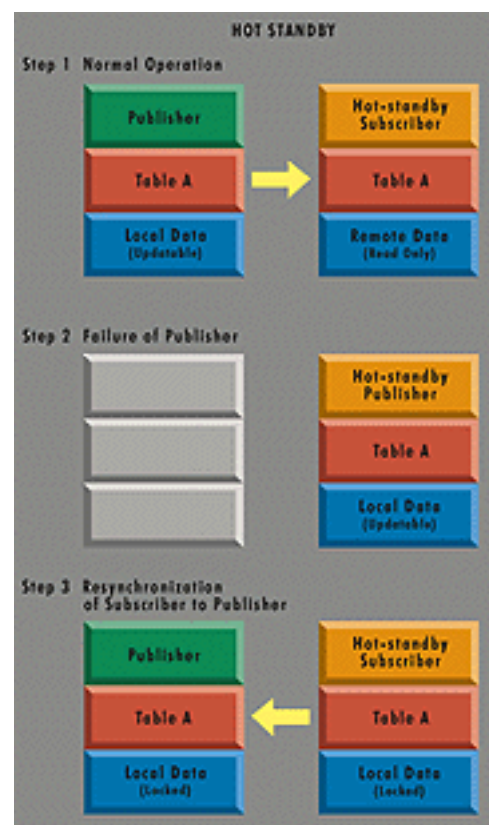
Building a Hot-Standby Solution

The UV/DR mechanism can be used to create hot-standby functionality with the Publisher being the main UniVerse server, and the Subscriber typically being the in-house development system or dormant fail-over server. In this mode, the frequency of replication is continuous, and again the Subscriber has a read-only copy of the source data. The Subscriber system is normally enabled with the Publisher's applications and replicated data directly from the Publisher system as well. Then, should the Publisher system fail, the Subscriber is ready to process the organization's business by itself until the original Publisher system is working again.

The UniVerse Admin tool handles the configuration of each Publisher and hot-standby Subscriber system from a networked PC. If the system does crash, UV/Admin manages the switch-over from the

Publisher to the hot-standby server. This occurs if the Publisher's disk integrity is compromised, the hardware is unusable, or in response to other failures that are beyond UniVerse normal warm-start recovery techniques. This methodology prevents the system switching over to the hot-standby system should there be a temporary interruption in the network connectivity between servers.

Figure 4. Stages of hot-standby operation using the UniVerse Data Replication mechanism.



When the hot-standby system is enabled, the Subscriber system acts as a fully-fledged Publisher and logs database updates. Previous target file replicas accept updates to the file. This operation continues until the Publisher system is ready again. Once this has occurred, UV/Admin freezes all updates on the Publisher and Subscriber systems and applies the replication log against the Publisher, if the original file was not lost. Having resynchronized both servers, UV/Admin switches the Publisher and Subscriber systems back to their previous roles.

Developers can easily create solutions that are a perfect match for an data access/replication task.

Product	Feature	Benefit
UV/Net	<ul style="list-style-type: none"> - Supports concurrent database access from remote UniVerse servers 	<ul style="list-style-type: none"> - Multiple servers can have read and write access to vital data - Easy to configure and administer, plus provides data consistency between sites - Exploits capabilities of standard TCP/IP networks
UV/DR	<ul style="list-style-type: none"> - Publisher system propagates copies of source data to one or many target Subscriber systems - This action can be configured for continuous, snap shot, or manually initiated mode 	<ul style="list-style-type: none"> - Less overhead than a truly distributed database - Greater degree of control of data access using read-only replicas - Exploits capabilities of standard TCP/IP networks

UV/DR in hot-standby mode	<ul style="list-style-type: none"> - Publisher system propagates a single copy of source data to a target hot-standby Subscriber system - Always runs in continuous mode 	<ul style="list-style-type: none"> - Quicker to get up and running than restoring from backup tapes - Effectively provides a continuous roll forward transaction log on the hot-standby machine - Increases robustness when used in partnership with other UniVerse and hardware solutions
---------------------------	--	---

Choosing the Right Solution

The availability of both UV/Net and UV/DR gives developers the ability to easily create solutions that are a perfect match for any data access/replication task. Table 1 provides an overview of the features and benefits of each solution.

Select UV/Net if you:

- want access to a single database for real-time, simultaneous, read/write access across multiple locations, e.g., a single central customer database, without changing your application code, or
- have a requirement to look at a complete picture of many separate inventory databases from a centralized location (i.e., corporate headquarters) for up-to-the-minute reporting purposes.

Choose Data Replication with UV/DR if you:

- are primarily interested in delivering non-updatable copies of your operational data without the need for sophisticated background tasks, or
- are only interested in last night's data, and want to minimize system load and network traffic by utilizing a single period of activity to perform batch mode updates.

Select UV/DR's hot-standby configuration if you:

- need to increase the robustness of the operational systems by providing a hot-standby application server with the latest copy of data, or
- want the additional security of a hot-standby solution even if disk mirroring and UniVerse warm-start recovery, or transaction logging mechanism, is being employed.

Technical Requirements

UV/Net

UV/Net works with any UniVerse release higher than 7.3.2 on UNIX, and 9.3.1 on Windows NT, in any release combination, using TCP/IP networking. It works seamlessly between each server with no programming changes required.

UV/DR and Hot Standby UniVerse

Data Replication and hot standby only function between UniVerse Release 9.4.1 or higher on UNIX, or Windows NT-based servers using TCP/IP networking.

UV/Admin must be used to configure both the Publisher and Subscriber systems. As it uses a lightweight logging mechanism, no programming changes or transactionalization of the application is necessary.

Availability and Packaging

Both UV/DR and UV/Net are part of the UniVerse Release 9.4.1 product. UV/DR is a no-charge feature of the 9.4.1 release. UNIX-based servers will require activation of the UV/Admin server component which is a separately-priced and authorizable product. UV/Net is optionally installable and a separately-priced product.