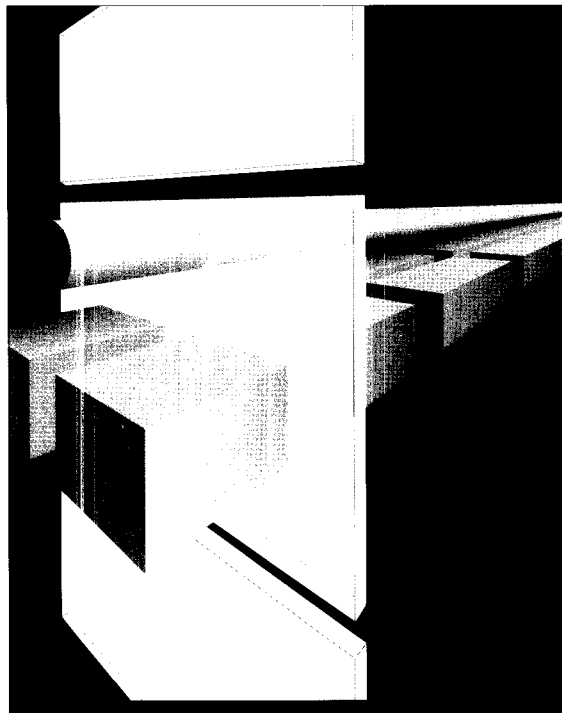


T A N D E M  
SYSTEMS REVIEW

---

VOLUME 8, NUMBER 1

SUMMER 1992



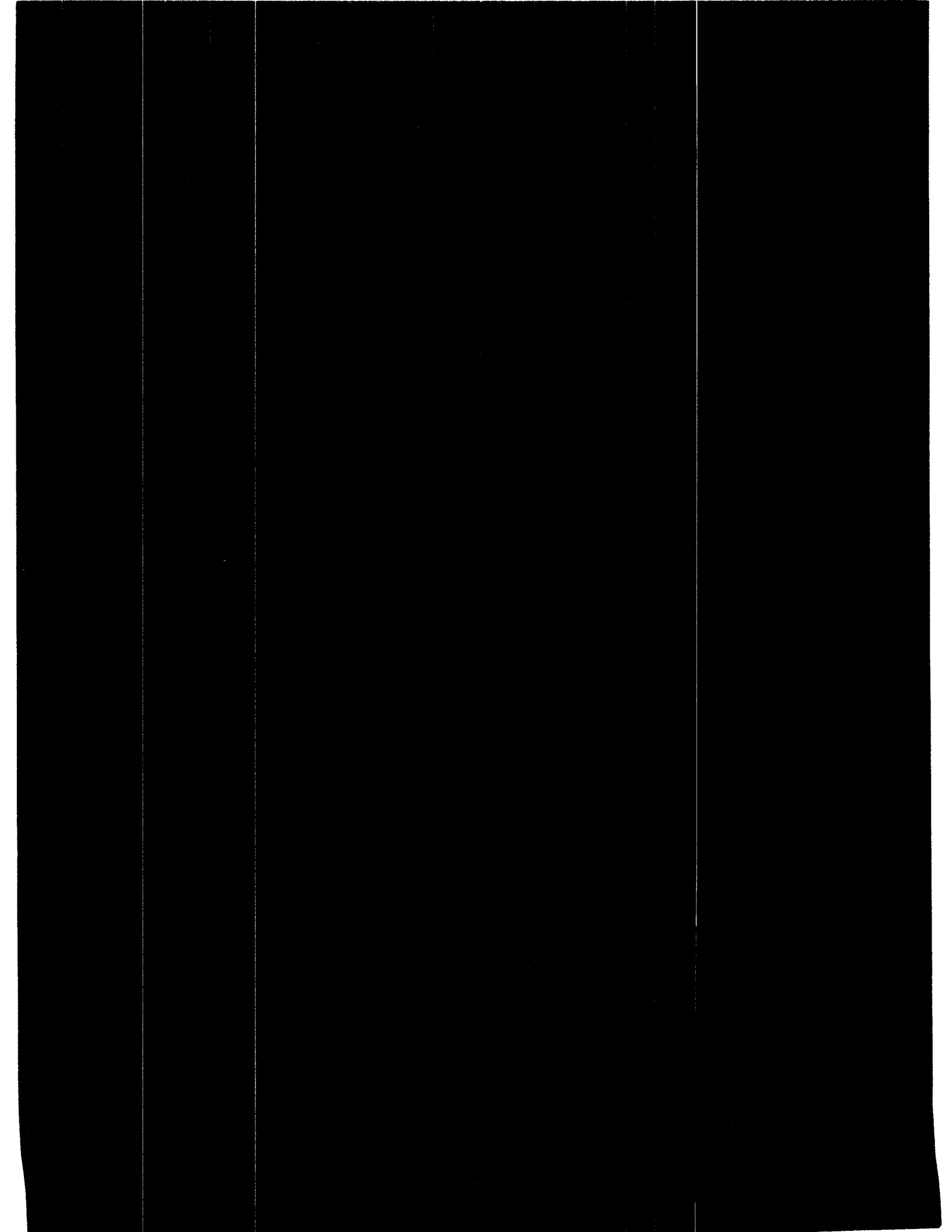
*RDF*

*Terminal Selection*

*Terminal Connectivity Options*

*Technical Information and Education*

*Product Update • Ongoing Support*



T A N D E M  
SYSTEMS REVIEW

VOLUME 8, NUMBER 2

SUMMER 1992





## Editor's Note

The number of alternatives for connecting terminals and workstations to host systems has increased dramatically as new standards and vendors have entered the marketplace. Tandem systems support many of these terminal connection alternatives while providing the flexibility and performance necessary to optimize applications. This issue includes two articles on terminal connectivity: "Terminal Selection" and "Connecting Terminals and Workstations to Guardian 90 Systems." By discussing and comparing the many terminal types and terminal connectivity options, these articles can help users determine the best design for their requirements.

The RDF (Remote Duplicate Database Facility) product makes it possible for a backup database on a remote system to take over database functions when the primary system is unavailable. The feature article in this issue, "RDF Synchronization," describes possible causes of desynchronization and methods of resynchronizing the databases.

This issue introduces a new department that will appear regularly in the *Tandem Systems Review*. "Technical Information and Education" is an annotated list of new consulting and information services, software education courses, and books that Tandem is offering its users.

— SWT

EDITORIAL DIRECTOR

Susan W. Thompson

EDITOR

Steven Kahn

ASSOCIATE EDITORS: David Gordon,  
Mark Peters, William Schlansky

ASSISTANT EDITOR: Hannah Silver

PRODUCTION MANAGER

Anne Lewis

ILLUSTRATION AND LAYOUT

Christine Kawashima

COVER ART: Steve Elwood

SUBSCRIPTIONS: Elaine Vaza-Kaczynski

ADVISORY BOARD

Mark Anderton, Jim Collins,  
Dennis Fukuyama, Terrye Kocher,  
Mike Noonan

*Tandem Systems Review* is published quarterly by Tandem Computers Incorporated. All correspondence and subscriptions should be addressed to *Tandem Systems Review*, 18922 Forge Drive, Loc 216-05, Cupertino, CA 95014.

Subscriptions: \$75.00 per year; single copies are \$20.00. Detailed subscription information is provided on the subscription order form at the end of this book.

Tandem Computers Incorporated assumes no responsibility for errors or omissions that may occur in this publication.

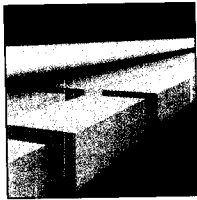
Copyright ©1992 Tandem Computers Incorporated. All rights reserved. No part of this document may be reproduced in any form, including photocopy or translation to another language, without the prior written consent of Tandem Computers Incorporated.

CLX, CLX/R, Cyclone/R, Envoy, Expand, Guardian, Guardian 90, Integrity, Integrity S2, Multilan, NonStop, RDF, SeeView, SNAX, Tandem, the Tandem logo, and TMF are trademarks and service marks of Tandem Computers Incorporated, protected through use and/or registration in the United States and many foreign countries.

Access/One, NIU, and Ungermann-Bass are trademarks of Ungermann-Bass, Inc.

UNIX is the registered trademark of UNIX System Laboratories, Inc. in the USA and other countries.

All brand names and product names are trademarks or registered trademarks of their respective companies.



## TERMINAL CONNECTIVITY

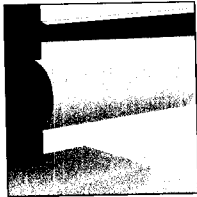
---

### **24** Terminal Selection

*Eric D. Siegel*

### **38** Connecting Terminals and Workstations to Guardian 90 Systems

*Eric D. Siegel*



## FEATURE

---

### **6** RDF Synchronization

*Wouter Senf, Frans Jongma*

## DEPARTMENTS

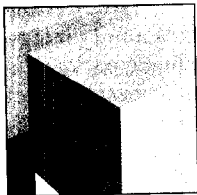
---

### **2** Product Update

### **60** Technical Information and Education

### **64** Ongoing Support: *InfoWay Online Support Service*

### **67** Index of Articles



## Guardian 90 Based Software

### Guardian 90 Operating System, Release C30.08

*July 1992*

The C30.08 release of the Guardian 90 operating system enhances the performance of Tandem's Transmission Control Protocol/Internet Protocol (TCP/IP) software, Expand networking software, and Exchange remote job entry (Exchange/RJE) software, as well as the Tandem group of OSI products. It also provides support for the 5200 Optical Storage Facility and the 5180 Cartridge Tape Subsystem on Cyclone/R systems and CLX 2000 systems.

### NonStop NET/MASTER Software *May 1992*

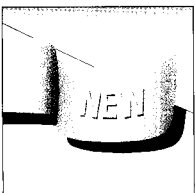
NonStop NET/MASTER software simplifies the management of Tandem NonStop systems. By automating many operations, it helps users reduce operations costs due to human error and time-consuming repetitive tasks.

NonStop NET/MASTER software, an extension of Tandem's Distributed Systems Management (DSM) architecture, allows users to cooperatively manage Tandem systems with IBM systems running NET/MASTER or NetView. By using NonStop NET/MASTER software, users can, from a single console, perform all critical management tasks on centralized or distributed systems and protect their systems from unauthorized access. Each console can be customized according to the user's needs.

### Information Engineering Facility for Tandem

*April 1992*

The Information Engineering Facility (IEF) for Tandem is an application development tool using integrated computer-aided software engineering (I-CASE) technology. IEF is an I-CASE product from Texas Instruments. It offers a fully integrated development software package, covering the entire application life cycle from information systems planning through ongoing application enhancement.



IEF for Tandem extends the range of the Texas Instruments product to Tandem's fault-tolerant NonStop architecture. It also incorporates a number of unique features, which include support for nonterminal devices, integration with Tandem operations-management tools, definition of NonStop SQL databases, automatic generation of online transaction processing (OLTP) test environments, and fault tolerance and data integrity.

With IEF, application design is independent of the target system. Developers using IEF can therefore focus on the business requirements of their applications rather than on considerations specific to a target system. Users of IEF for Tandem can quickly create, in either C or COBOL, complete OLTP and batch applications for Tandem NonStop systems.

### **Two-Processor Cyclone/R System**

*April 1992*

The Cyclone/R system is now available in a two-processor configuration (G5020) that extends the Cyclone/R system family to 27 transactions per second (TPS) per system. The two-processor Cyclone/R system provides a lower entry point for the Cyclone/R system family, making it easier for users to migrate to RISC processing.

### **Improved Disk Drive for the XL80 Disk Storage Facility**

*February 1992*

A new disk drive is now available for use in the XL80 disk storage facility. The new drive provides improved reliability with price and performance similar to that of the original disk drives.

### **Remote Server Call**

*November 1991*

The Tandem Remote Server Call (RSC) provides the means to connect PCs with Pathway and non-Pathway servers. By integrating PCs and other intelligent devices with Tandem's Guardian 90 and Pathway systems, RSC allows users to implement OLTP applications based on a client-server architecture. The PC interface in RSC supports most PC-based languages, 4GLs, tools, and other applications.

RSC optimizes desktop and main-frame computing resources for OLTP. RSC is especially beneficial to users who have Pathway-based applications, allowing PC-based clients to take full advantage of their servers without changing existing server code.

### **Tandem SQL Server Gateway**

*December 1991*

Tandem SQL Server Gateway makes it possible for applications and tools supported by SQL Server to access NonStop SQL databases through Tandem's relational database management system (RDBMS), NonStop SQL. SQL Server is an RDBMS that runs on OS/2 and UNIX workstations. Tandem SQL Server Gateway runs on a Tandem Guardian 90 host.

Tandem SQL Server Gateway uses a client-server architecture, in which a client-based application process makes calls to a server-based database process. Clients are LAN-based (PC or workstation) applications and tools that use the SQL Server interface. Clients communicate with Tandem SQL Server Gateway as if it were an actual SQL Server.

### **3602, 3603, Envoy on RISC Systems**

*March 1992*

The 3602 and 3603 controllers and Envoy data communications manager are now available on the Cyclone/R system and on all models of the CLX/R system, as well as on the CLX 2000 system. To use these products on their RISC systems, users must have the C30.07, or later, release of the Guardian 90 operating system. This release of the Guardian 90 system includes Tandem Maintenance and Diagnostic System (TMDS) support for the 3602, 3603, and Envoy products.

### **OSI/MHS Gateway Programmatic Interface**

*March 1992*

The OSI/MHS Gateway Programmatic Interface (GPI) provides users with a gateway, based on the X/Open and X.400 API Association (XAPIA) standards, that allows programmatic access to Tandem's OSI/MHS product. The OSI/MHS GPI provides access for proprietary applications interfacing to the X.400 message-handling system from a Guardian 90 operating system environment. The OSI/MHS GPI is now available to all users.

### **NonStop-UX Based Software**

### **ORACLE Relational Database Management System, Release 6.0.33**

*March 1992*

Release 6.0.33 is Tandem's latest release of the ORACLE relational database management system (RDBMS) and suite of tools for the Tandem Integrity S2 systems. The new release provides transaction-level control of rollback segment use. It also makes available a new version, release 5.011, of SQL\*Menu. This release of SQL\*Menu takes advantage of Oracle's standard menu-handling and terminal interface.



## **Informix Relational Database Management System, Release 4.1**

*February 1992*

Release 4.1 of the Informix relational database management system (RDBMS) operates with either of two, separately available database engines, called Online and Standard Engine. Online is the second-generation online transaction processing (OLTP) engine from Informix. It replaces, and is upwardly compatible with, the previous version, Turbo. All previous versions of Informix products are compatible with the release 4.1 engines.

The Informix Online system offers the following new features:

- Multimedia support (BLOBS) with the Online engine.
- A cost-based optimizer that reduces the time needed to perform complex queries.
- Advanced logging, checkpoint, and recovery schemes that automatically recover the database if a system interruption occurs.
- Access of data from geographically dispersed databases.

## **INGRES Relational Database Management System, Release 6.3**

*May 1992*

Release 6.3 of the INGRES relational database management system (RDBMS) supports distributed database processing and client-server architectures. INGRES/Star, a module of the INGRES 6.3 RDBMS, creates a single, unified information resource from data residing in virtually any combination of local or remote INGRES databases.

The INGRES 6.3 database manager is optimized for handling OLTP applications. A multithreaded-server feature enables the database manager to handle multiple requests simultaneously. The system also provides an online-backup feature. Where systems run 24 hours a day, 7 days a week, this feature permits users to perform periodic system maintenance without affecting normal operations.

The two-phase commit process of INGRES 6.3 ensures the integrity of distributed transactions that span multiple computers. In addition, the multivolume-table feature makes it possible to create large data tables and to distribute them among multiple disk volumes.

## RDF Synchronization

**T**he Tandem™ RDF™ (Remote Duplicate Database Facility) product makes it possible for a backup database on a remote system to take over database functions when the primary system is not available. This protects data and provides maximum continuity of online processing in sudden emergencies or when a planned shutdown of the primary system is necessary.

For the backup database to successfully take over database operations, it must be an accurate copy of the primary database. The two databases must be synchronized. However, certain exceptional conditions can desynchronize the databases. When this happens, it is important to resynchronize the databases as quickly as possible.

This article describes possible causes of desynchronization, ways of preventing desynchronization, and methods for resynchronizing databases. Resynchronization can be carried out both online and offline. Online resynchronization allows the database application to remain fully operational. Offline synchronization is often faster but requires the application to be halted temporarily.

The following material assumes that the reader is familiar with RDF. The content applies only to the use of RDF with release C30 of the Guardian™ 90 operating system. Throughout the text, *table* refers to a table created through Tandem's NonStop™ SQL relational database management system; *file* can refer either to a file created under Tandem's Enscribe record management system or to a file created through NonStop SQL. Under NonStop SQL, files and tables are equivalent objects.

Figure 1

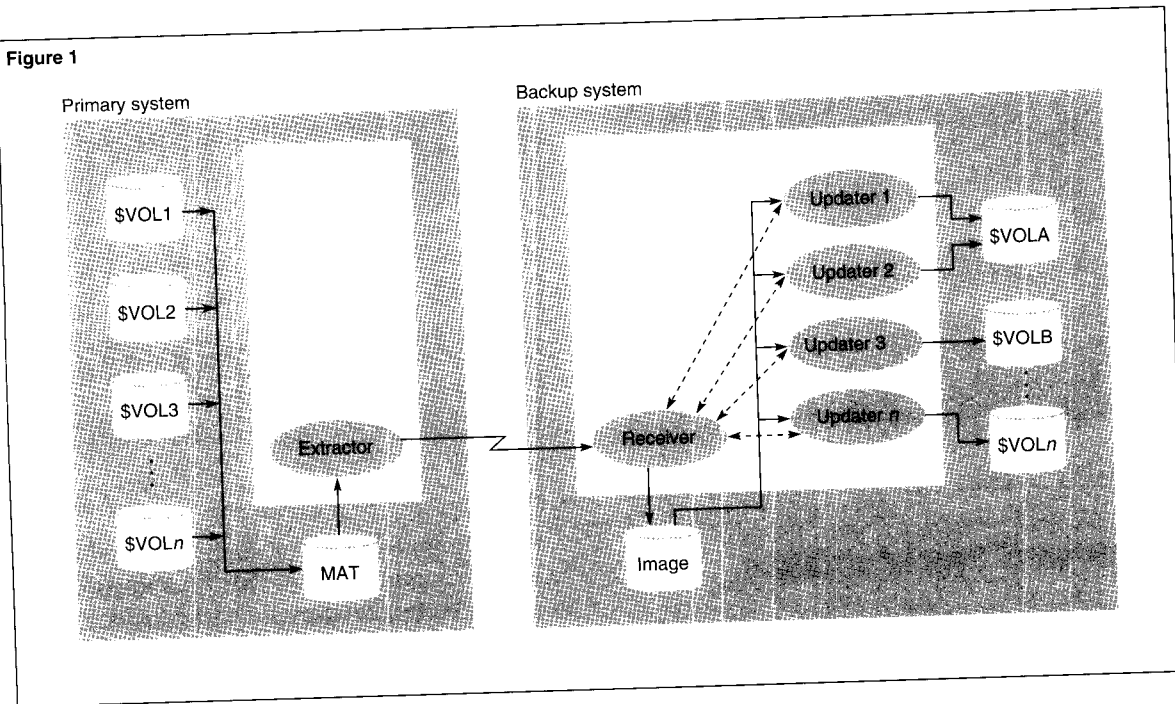


Figure 1.  
Basic RDF configuration.

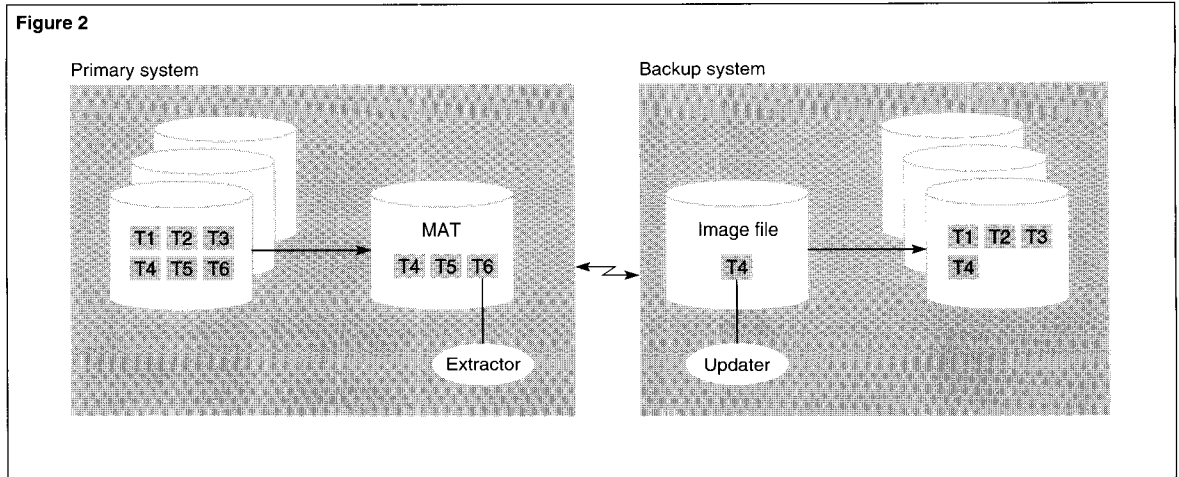
## Basic Features of RDF

RDF replicates a primary database on a backup system by detecting transactions that change the primary database (inserts, updates, and deletes) and applying the changes to the backup database. To do this, RDF reads a log of the database transactions carried out on the primary system and transmits it to the backup system. The log is maintained by Tandem's TMF™ (Transaction Monitoring Facility) software in the Master Audit Trail (MAT) file. When MAT data arrives at the backup system, it is stored in RDF image files. RDF updater processes use the image files to update the backup database. Since RDF transfers only TMF data, the backup system maintains data only for tables that are audited on the primary system.

Figure 1 illustrates a basic RDF configuration for maintaining backup and primary databases. It will be the basis for all subsequent discussion, although other, more complicated, configurations are possible. For more information on RDF, see "RDF: An Overview" in the October 1991 issue of the *Tandem Systems Review* (Guerrero).

Figure 1 shows audited volumes on the primary database as the source of MAT data. An extractor process extracts MAT data from volumes configured under RDF. This data is transmitted over a Tandem Expand™ data communications network to a receiver process. The receiver process relays the MAT data to the backup system's image disk and communicates with updater processes on the backup system.

**Figure 2.**  
Synchronized databases  
during RDF operations.



## Definition of Synchronization

Two databases are synchronized if either (a) they both have identical contents, in terms of data configured under RDF on the primary system, or (b) in the course of its normal functioning, RDF can make up for any difference between the two databases, as in the case of a time lag between data arriving in the MAT file and being transmitted by RDF. This is equivalent to the definition provided in Appendix I, "Supplemental Information for Release C30," of the *Remote Duplicate Database Facility System Management Manual* (1990/1991).

Figure 2 shows synchronized primary and backup databases. Although the backup database has received only four of the six transactions carried out on the primary system, RDF is functioning normally, and transactions T5 and T6 are available for transmission to the backup system.

If communication over the Expand network fails in a situation such as that illustrated in Figure 2, both databases are still considered synchronous, even though a large amount of untransmitted data may accumulate on the primary system. This follows from the second part of the definition of synchrony. Once communication between systems is restored, RDF can resume transmitting data from its previous location in the MAT file. Aside from the delay in transmission, the databases are equivalent.

Even though synchrony is preserved during a prolonged communication break, one might decide to resynchronize the databases using one of the offline methods described later. Resynchronization, in this case, minimizes the possibility that a failure on the primary system will require the backup system to take over operations while missing a substantial amount of data.

If there is untransmitted MAT data on the primary system, a failure on either system can desynchronize the databases. This is because RDF maintains contextual information on both systems that is necessary for extracting or receiving data. If either system fails, contextual data may be lost. Without complete contextual data, even if both systems return to operation, RDF cannot identify the transactions on the primary system that still need to be sent to the backup system. In this state, RDF stops operating on both systems.

---

## Causes of Desynchronization

Loss of synchrony between primary and backup systems is uncommon. RDF is designed to handle the great majority of disruptions, including

- Data communication errors.
- Planned shutdowns of either the primary or backup systems (as may be necessary for upgrades to new releases of the Guardian 90 operating system).
- Most unplanned or emergency shutdowns of the primary system.

A failure of the primary system does not cause desynchronization as long as RDF has already sent the most recent MAT data to the backup system. In general, when desynchronization does occur, the primary causes are the following:

- A failure on the backup system results in lost data or causes the RDF context to be lost.
- A sudden failure on the primary system prevents RDF from transmitting the last few MAT updates to the backup system.
- The location or layout of tables on the primary system is changed without corresponding changes on the backup system.
- Operator error, such as purging a required file, destroys data on the primary or backup system.

A failure on the backup system can cause RDF to lose the context it maintains for monitoring incoming MAT data. As described earlier, this desynchronizes the databases and stops the primary system from sending further MAT data, making resynchronization a necessity.

A sudden failure on the primary system can prevent RDF from sending the last few MAT updates or commit-abort status records to the backup system. When this happens, the backup system may need to take over operations with a slight asynchrony. In some cases, user-developed software can restore the missing updates (see "Recovering Updates Not Transmitted During a System Failure").

RDF does not transmit structural database changes or changes in the location of files. If users make either type of change on the primary system, the backup site must make corresponding changes in order to preserve synchrony.

---

## Preventing Desynchronization

Preventing desynchronization should be a high priority. If the primary and backup databases are out of synchrony while the database application is active, a failure of the primary system may force the backup system to take over operations while missing an important body of transactions. In addition, resynchronizing databases can be a lengthy process. If it is carried out offline, the database application is inactive. If it is carried out online, the process takes longer and the application is active while the databases are out of synchrony.

### Use Mirrored Disks

Mirrored disks preserve data. If data is important enough to save through RDF, it is probably important enough to save on mirrored disks. It is highly recommended that all disks configured under RDF, on both primary and backup systems, be mirrored.

If data is not stored on mirrored disks, every disk failure desynchronizes the databases. This can result in time-consuming resynchronizations. Suppose, for example, that  $f$  is the average number of disk failures per disk per year, and that there are  $n$  unmirrored volumes on the primary system and  $m$  unmirrored volumes on the backup system. This means there will be  $f \times n$  disk failures on the primary system, requiring  $f \times n$  takeovers by the backup system. There will be  $f \times m$  failures on the backup system. Altogether, resynchronization will be necessary  $f(m+n)$  times a year. The more unmirrored disks there are on each system, the greater the risk that overlapping disk failures on the two systems may completely halt the application.

If mirroring all disks on both systems is not feasible, an alternative is to mirror all disks configured for RDF on the primary system and to mirror at least the image disk on the backup system. If the image disk is unmirrored and fails, the backup system must be completely resynchronized. If a non-image disk on the backup system fails, only its contents, not the entire database, must be resynchronized. A full resynchronization of the entire database is not necessary. Resynchronizing a single disk is made considerably easier if an installation configures RDF so that disks on the primary and backup systems are in one-to-one correspondence.

## **Make Structural Changes on Both Systems**

RDF replicates changes in the content of a database, but it does not transmit structural changes such as the creation of tables and indexes or the addition of new columns to a table. As a result, if users make structural changes to the primary database, the backup site must make the same changes on the backup system. Without corresponding changes, updates to new structures on the primary system have no place to go on the backup system and the databases are immediately out of synchrony.

### ***Creating New Tables, Columns, and Enscribe Files.***

To maintain continuous operation of the database application on the primary system, first add new tables, columns, or Enscribe files to the backup database, then add them to the primary database. This ensures that any updates to the new objects on the primary database have a place to go on the backup database. If the database application is stopped or new objects are created with read-only access, users can add an object to either system first. The only requirement is that an application not make updates to an object on the primary system before the corresponding object exists on the backup system.

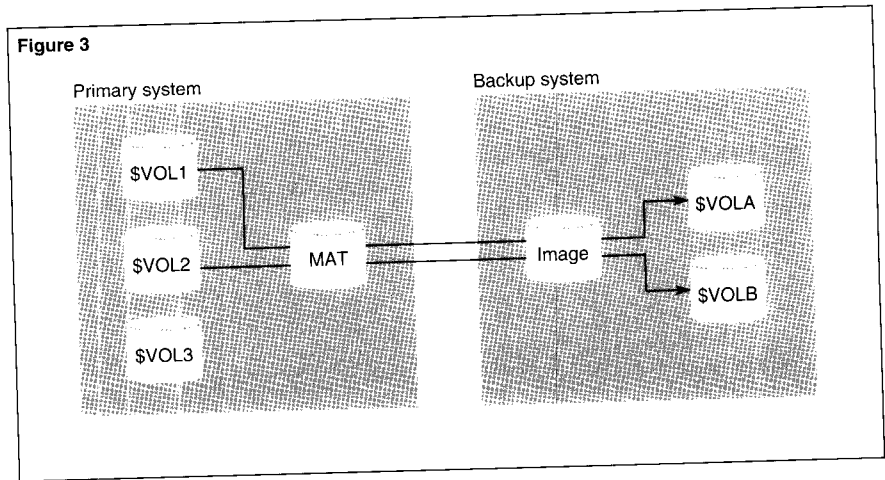
***Adding Indexes.*** A user can add an index to the primary database by stopping updates to the base table or file (read-only access is allowed) and creating the index. The corresponding index can be added to the backup database only after updates to the primary table have stopped and RDF has brought the table or file on the backup system completely up to date.

The two databases build indexes independently. As long as the base tables or files start off with identical contents, the order in which the indexes are created does not matter. The two databases can build corresponding indexes simultaneously. If the base table on the primary system receives updates while the backup database builds an index, RDF must run in Update Off mode. When the index is completed, RDF can again run with Update On.

**Adding Views and Table Constraints.** One can create views and table constraints on either database first. The absence of a view or constraint on the backup system does not affect data synchronization. However, as soon as a view or constraint is created on the primary system, it should be added to the backup system. When there is a switchover and the database application runs on the backup system, the application is likely to make use of the added view or constraint.

**Deleting Tables, Enscribe Files, and Indexes.** When a table, Enscribe file, or index is removed from the primary system, it should also be removed from the backup system. If the corresponding object is left on the backup system, it takes up unnecessary disk space. It can also cause problems in the event that a user creates a new audited object on the primary system with the same name but with different contents.

Serious problems may arise if an index is deleted on the primary system but not on the backup system. After a takeover, programs that use indexes to retrieve data may find the undeleted index and return invalid data or abort while trying to access a record that no longer exists.



### Perform a Switchover Before Upgrading to New Disks

Occasionally one needs to change the initial configuration of files and disks on the primary system. For example, as smaller disks start to fill up, a site may decide to upgrade to larger disks. By allowing a switchover to the backup system, RDF makes it possible to preserve synchrony and keep the database application running while the audited files on one disk are moved to another disk. The basic procedure for moving the files is illustrated in Figures 3, 4, and 5. Initially, the primary system uses two disks, \$VOL1 and \$VOL2, to store audited files. The goal is to move all audited files from \$VOL2 to a new disk, \$VOL3. Figure 3 shows the initial configuration under RDF.

In Figure 3, RDF replicates \$VOL1 and \$VOL2 on \$VOLA and \$VOLB of the backup system. \$VOL3 is a new disk, currently not configured under RDF.

Figure 3. Configuration before moving files.

Figure 4

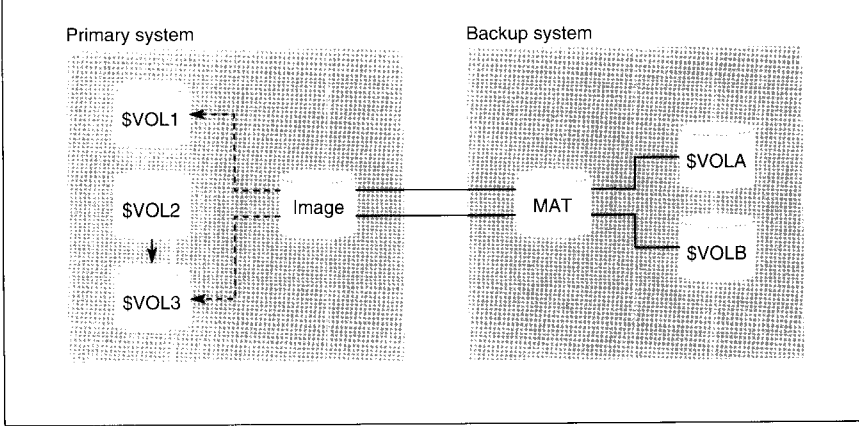


Figure 4 illustrates a switchover to the backup system and the transfer of files from \$VOL2 to \$VOL3 on the primary system. RDF must run in Update Off mode during the file transfer.

In Figure 4, RDF on the backup system is reconfigured so that data from \$VOLB goes to \$VOL3 instead of \$VOL2. As illustrated, the backup system sends audited data from disks \$VOLA and \$VOLB to its own MAT file. From there, RDF transmits updates to an image file on the primary system. The RDF updaters (dotted arrows) on the primary system must be disabled so that audited files on \$VOL2 can be moved to \$VOL3.

Once \$VOL3 has been configured into RDF and the transfer of files from \$VOL2 to \$VOL3 is complete, one can activate the updaters on the primary system to bring both \$VOL1 and \$VOL3 up to date. Finally, as shown in Figure 5, a switchover returns operation of the database application to the primary system. Audited data from \$VOL3 now goes to \$VOLB on the backup system.

Figure 5

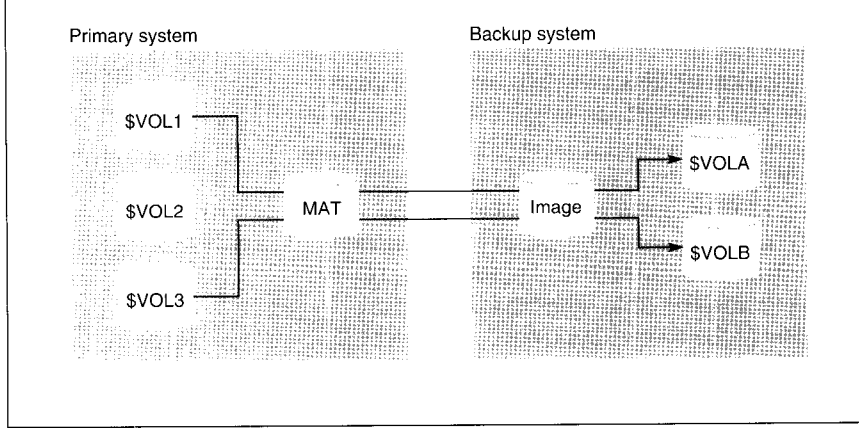


Figure 4.

Backup system takeover while primary system moves files.

Figure 5.

RDF configuration after transfer of files.

### Install Guardian 90 Upgrades Using the Old Release as Backup

RDF can contribute to safe upgrades of the Guardian 90 operating system. A number of different upgrade approaches using RDF are available. The choice of a particular approach depends on such factors as whether

- There is a separate development system for testing.
- The primary and backup systems are the same size and configured in the same way.
- One system is favored over the other for production purposes.

Whichever approach an installation chooses, when the new release is first used in the production environment, the old release should still be available on the system that serves as the backup.



Before upgrading a production system to a new Guardian 90 release, the release should be thoroughly tested on a development system. However, even the most thorough testing on a development system cannot cover all the possibilities of a production environment. RDF contributes to safe upgrades by making it possible, after initial testing, to run the new release in a production environment on one system, while running the old release on the other system. In this way, if the production system fails, whether because of the new release or for other reasons, the backup system can reliably take over with the old release. Once a new release has demonstrated sufficient reliability under production conditions, it can be loaded on the system serving as a backup.

### **Run New Database Application Releases With RDF in Update Off Mode**

In contrast to Guardian 90 release upgrades, only one version of the database application can be active. If the application is running on the primary database, it cannot also be executing on the backup database.

After initial testing, whether on an independent development system or on the backup system, the safest procedure is to load the new application release on the primary system and run RDF in Update Off mode. In this way, if the application quickly fails under production conditions, RDF has not replicated corrupted updates on the backup system. Once it is clear that the application functions satisfactorily, RDF can be returned to Update On mode. As a precaution, one should be ready to stop the application on the primary system and either immediately reinstall the old version of the application on the primary system or perform a takeover with the old version on the backup system.

### **Make Sure That Required Audit Flags Are Turned On**

It is important for both the backup and primary sites to have an up-to-date list (online and hard copy) of all files that need to be audited. The most reliable way of maintaining such a list is to develop a program that generates new lists on a regular basis.

RDF transmits only audited data. If the audit flag for a required file is not on, RDF cannot transmit updates to the file, and the databases will get out of synchrony. This problem can arise in connection with a planned or unplanned takeover.

Audit flags on the system running the database application must be turned on. Audit flags on the backup system receiving updates through RDF must be turned off. This is usually not a problem until an initial takeover is necessary. During a takeover, the application is started on the backup database, and audit flags for the required files on the backup system must be turned on. At this point the primary system may not be in operation. However, as soon as it is available, audit flags on the primary system must be turned off, so that RDF can run in reverse and update the primary database with changes made to the backup database after the takeover. This resynchronizes the contents of the two databases so that the primary system can resume execution of the database application. Finally, when the primary system takes back the application, its audit flags must be turned back on, and the audit flags on the backup system must be turned back off.

If the backup and primary sites do not maintain current and complete lists of files to be audited, some audit flags may not get turned on when the backup system performs a takeover. Similarly, some audit flags may be missed when the primary system resumes operation of the database application. The application itself is not likely to recognize that required audit flags are not turned on. As a result, the databases will lose synchrony and require resynchronization. Attention should also be given to turning on audit flags after creating and loading a new file on the primary system. Loading a file is typically an unaudited function; once the file is loaded, its audit flag must be turned on.

---

## Monitoring Synchronization

Loss of synchrony between databases is not always obvious. As a result, users should monitor databases regularly to detect desynchronization as early as possible. In addition, synchronization should be checked whenever RDF updaters on the backup system report unexpected RDF error messages for records that already exist (700 File System Error 10) or records that are not found (700 File System Error 11); these are prime indicators of desynchronization.

RDFCHEK, a file-comparison program provided with RDF, can check database synchrony if no updates are carried out while it runs. Before RDFCHEK can run, the database application must stop making updates and RDF must catch up with the MAT data already generated.

To make reliable checks of synchrony possible without stopping updates to the database, the database application must maintain either a last-update timestamp for each record or assign consecutive update sequence numbers. This allows user-developed monitoring software to scan tables or Enscribe files on the two databases and identify individual transactions, or ranges of transactions, that are missing from one of the databases.

If the primary database is operated continuously, the primary and backup databases may never be logically identical, since there is a lag between the time a transaction is committed on the primary database and the time it is replicated on the backup database. The extent of the time lag depends on how great a backlog individual RDF updaters on the backup system have; some updaters may be further behind than others. If an application monitoring synchrony finds that the backup database is missing updates carried out on the primary database, it must allow enough time for the RDF updaters to make up the time lag and then recheck the missing updates.

Since each disk volume on the primary system has a single updater on the backup system, the contents of a single file or table partition on the backup system should be in one-to-one correspondence with the contents of the same file or partition on the primary system at an earlier point in time. The following example illustrates one of the many possible approaches to testing for synchrony between database files. The example uses a file with timestamped records.

On the backup database, establish a reference time by finding the latest timestamp entered in the file. Suppose this is exactly 12 noon of the current day. Compare all records in the backup file and primary database file with a timestamp equal to or earlier than 12 noon. All records in the file on the primary system with a timestamp of 12 noon or earlier should have duplicates in the file on the backup system. If one or more records in the primary system file are not replicated in the file on the backup system, the databases are out of synchrony.

---

***M***onitor databases regularly to detect desynchronization as early as possible.

If a record in the file on the backup system does not have a duplicate on the primary system, it may not mean that the files are out of synchrony. It is possible that the corresponding record on the primary system was deleted or updated after the specified reference time and the audit from this transaction has not yet been applied to the file on the backup system. To evaluate whether this is the case, wait long enough for the updater responsible for the file to apply the update or deletion. At that time, if the record is not updated or deleted, the databases are out of synchrony.

If the backup and primary databases are large, a full comparison of all files or partitions may be too time consuming. In this case, user-developed software can apply sampling techniques to evaluate the likelihood that the databases are out of synchrony.

Because each updater runs at its own rate, the contents of the backup database as a whole may not be the same, at a given time, as the contents of the primary database at any earlier point in time. As a result, if one wants to select a single reference time for use in making all file comparisons throughout the database, the time must be based on the RDF Time Delay (RTD) of the slowest updater. One can display RTDs for all updaters with the STATUS RDF command under RDFCOM.

For example, suppose all RDF updaters except \$UPDATE4 have an RTD of less than 2 minutes. \$UPDATE4 has an RTD of 15 minutes and 30 seconds. A fixed reference time for all database file comparisons cannot be more recent than 15 minutes and 30 seconds ago.

---

## Resynchronization Methods

There are a number of methods for resynchronizing databases. Online methods allow the database application to run continuously during resynchronization. Offline methods require temporarily stopping the application, but are often considerably faster than online methods. The two types of methods can also be combined.

The choice of a specific method, or combination of methods, is likely to depend on a variety of factors. These include:

- The cause of desynchronization.
- The availability of the primary system.
- The necessity of maintaining continuous operations.
- The size of the database.
- The extent of the desynchronization.
- The risk involved in running the database application without a backup system.

For example, if only a few updates are lost because a sudden failure on the primary system prevented RDF from sending the last few completed transactions, specially written software may be able to recover the missing updates. Once recovered, the updates can be applied to the backup system either manually or programmatically.

If a full resynchronization is necessary and keeping the database application in operation is important, one can often carry out the resynchronization using RDF in conjunction with user-developed programs. If stopping the

application is feasible, one can copy the desynchronized files or the entire database onto tape, and then

use the tape to restore the copied data on the desynchronized system. Other methods of resynchronizing the databases can also be used in this situation.

Resynchronization can be a time-consuming and expensive process, and it may not be called for in all cases of desynchronization. For example, suppose that 15 records are missing from a large file on the backup database, and each record is worth three dollars. The value of having the records in the database may not be worth the expense of resynchronizing the entire file. It may or may not be worth the effort of inserting the records manually.

Resynchronization may also not be necessary if records are missing from a file or table whose records get aged and deleted after a short time. As soon as the lifetime of the missing records has been exceeded, they will be deleted on the primary system, and the contents of the two databases will be back in synchrony. RDF sends the deletions to the backup system, where they generate RDF error messages for unfound records (700 File System Error 11). The errors are recorded in the RDF error log.

Resynchronization is a planned activity. Each RDF site should have instructions available in advance, both online and on paper, for each type of resynchronization. The methods may be described as part of a disaster recovery plan or business contingency plan.

---

## **Recovering Updates Not Transmitted During a System Failure**

If the primary system suddenly fails and the backup system must carry out a takeover, last-minute updates completed on the primary system may not reach the backup system. Without the updates, the databases are desynchronized, and the database application has to run on the backup system with data missing. In some cases, user-developed software can recover the missing updates.

### **Client Systems Resend Missing Updates**

If intelligent clients (systems with their own CPUs and memory) initiate transactions, a site can write software to have the clients resend missing updates directly to the backup system. Part of the software must reside on the client, part on the Tandem system running the application. Under the software on the client, each client system maintains a history of the transactions it sends and records whether or not the transactions were acknowledged as completed. All transactions sent by an individual client identify the sender and are sequentially numbered. On the primary system, the software maintains a log file containing the transaction numbers of all completed transactions. This file is protected by TMF and replicated on the backup system by RDF.

---

## ***Resynchronization is a planned activity.***

After a takeover, clients send transactions to the backup system. The user-developed software must make possible the following scenario. When the backup system receives an update from a client, it checks the transaction log file received from the primary system to make sure the current transaction number immediately follows the previously highest transaction number from the same client. If it does, the system processes the transaction and sends back an acknowledgment of the completed transaction. If the current transaction number is too high, it means one or more transactions sent to the primary system are missing. In response, the backup system sends a message to the client system asking it to resend the missing transactions. The client system receives this message, refers to its transaction history, and resends the missed transactions.

### **Primary System Sends Missing Updates**

If terminals, rather than intelligent clients, send transactions to the primary system, and the primary system is capable of recovery with no loss of data, software on the primary and backup systems may be able to handle missing updates. Users can design the software in a number of ways. One approach is to program the following scenario.

As soon as the primary system is available, the backup system searches the backup transaction log file for the number of the last committed transaction received before the primary system's failure. The backup system sends the number to the primary system. The primary system uses the transaction number to search through either a user transaction log file or the MAT to find all transactions completed after the backup system's last transaction. These transactions are sent to the backup system. User-developed software on the backup system receives the transactions and updates the appropriate files or tables.

By leaving out the initial steps, one can simplify the preceding scenario, as follows. Software on the primary system reads the transaction log file and sends update transactions to the backup system, starting at a point before

the primary system's failure. The selected time point is early enough to include all updates missing as a result of the failure. User-developed software on the backup system receives the transactions and carries out the updates. Resent transactions that were received before the primary system's failure will generate error messages. In many, but not all, cases, the messages can be ignored. To examine the cause of the messages, the user-developed software on the backup system can generate an exception report listing resent transactions that caused error messages or were in conflict with existing record data.

---

## **Resynchronizing During Continuous Operation of the Application**

There are two basic approaches to carrying out a resynchronization while the database application is online. Both approaches require user-developed software.

In the first approach, a user-developed program copies files from the functioning database while RDF operates in Update Off mode. In the second approach, a user-developed program deletes and reinserts records into the functioning database while RDF transmits these and all other updates to an empty database on the system being restored. Each of the approaches has benefits and drawbacks. In some cases, a combination of both methods may be the best solution.

Advantages of the first approach, using a copy program, are that it does not significantly add to contention for database resources and can be used with all types of structured files. A disadvantage is that RDF must run in Update Off mode, which can create disk-space problems as audit data accumulates in image files on the backup system. Because of this, it is usually desirable for the copy process to finish in a relatively short time.

An advantage of the second method, using a delete-reinsert program, is that RDF can run in Update On mode. This prevents disk space for image files from becoming a problem and

allows more time for resynchronization.

However, this approach may not be suitable for some databases with

audited entry-sequenced files. In addition, it may increase database contention.

Neither approach should be used with an Enscribe database if the application can delete records from a file or update records in an index file during resynchronization. As explained below, if the system rolls back transactions containing either type of operation, the resynchronization methods described here may fail to copy valid records.

***Each approach has its own advantages.***

Under NonStop SQL, when a record is to be deleted, a deletion lock is placed on the record that physically follows it. The deletion lock prevents scans and updates belonging to other transactions from proceeding past the deleted record until after the delete transaction is committed. In this way, if the delete transaction is aborted, TMF can back out the incomplete transaction, and the record involved can be included in the waiting transactions.

Under Enscribe, when a record is to be deleted, no lock is held at the position of the next record. Thus, if either a user-developed copy program or delete-reinsert program is operating on an Enscribe file from which a record has been deleted, it can move on to records following the deletion. If the system backs out the delete transaction, the program may already have passed beyond the restored record, and it will never be copied. Under Enscribe, when an index record is updated, the entire record is deleted and replaced with the updated version. This creates the same problem for user-developed programs as a simple delete operation.

### **User-Developed Program Copies Files**

Under this approach, a user-developed program copies database files while the application is running. When contention for database resources is a concern, it is the preferred method. The general procedure is:

1. Start RDF in Update Off mode.
2. Copy database files directly to the remote system across the Expand network, or copy the files to tape and transport the tape to the remote system.
3. When copies of all files are installed, put RDF in Update On mode to bring the remote database up to date.

As an intermediate step, it may be useful to copy database files to a separate disk rather than immediately transmitting them over Expand or copying them to tape. The disk copy can then be transmitted over the Expand network or backed up to tape. In this way, if a problem arises at some point during transmission over Expand or in restoring the tape at the backup system, the original copy is still available on disk.

To avoid duplicating uncommitted data, the copy program must use Stable Access for scanning tables in NonStop SQL databases. For files in Enscribe databases, it must not use the read-through-lock option. In addition, the program cannot use methods that ignore locks. It cannot use the FUP COPY, FUP DUP, SQLCI COPY, or BACKUP commands. All of these commands ignore locks and can copy uncommitted data.

The copy program executes with RDF in Update Off mode. Once the copy is on the backup system, RDF is put in Update On mode. As the RDF updaters start applying audit from the image files on the backup system, they may begin to report frequent RDF error messages for records that already exist (700 File System Error 10). These messages are generated because the copy program may copy records that were inserted after it started. Frequent error messages for records not found in a file (700 File System Error 11) may occur because the copy program missed records that were deleted after it started. If it is feasible to suspend all outside delete and insert transactions initiated while the delete-reinsert program is in operation, the error messages can be completely avoided.

Once all RDF updaters have passed the point at which the copy program finished executing, RDF file system errors should stop. To monitor the progress made by the updaters, an operator can use the RDFCOM STATUS RDF command to check delay times.

While the user program copies the database and the copy is installed on the backup system, RDF continues to transmit MAT data from the primary system to the image disk on the backup system. The image disk must not be allowed to fill up before installation of the copy is complete and the RDF updaters can be turned on.

If the image disk fills, RDF stops sending data to the backup system. As a result, the MAT disk on the primary system starts to fill up. If the MAT disk reaches the MAXFILES limit, TMF stops accepting new audit data. Once this happens, the database application can no longer update audited files and may be brought to a halt.

To prevent the image disk from filling, users can back up image files as the image disk fills and periodically purge the files from the disk in order to create more space. Once the copy of the active database is installed, the backed-up image files can be restored and the RDF updaters reactivated.

When possible, an installation should usually carry out an online resynchronization with a copy program during off-peak hours. Because of reduced contention between the copy program and the database application, this can both speed up the resynchronization and improve system response for application users. In addition, because the application generates less audit data during off hours, image files on the backup system will not fill up as rapidly, and there may be more time for copying the database before space on the image disk becomes a concern. Finally, if the copy is sent directly over an Expand network, the application's lowered transaction rate will leave more bandwidth free for the resynchronization.

### **User-Developed Programs Delete and Reinsert Records**

In this approach, user-developed programs delete and reinsert records into files in a functioning database while RDF runs in Update On mode. RDF transmits the deletions and reinsertions to an empty database on the backup system.

The delete-reinsert programs must apply to all audited key-sequenced and relative files. Such programs can also be extended to entry-sequenced files, but it is not always practical to do so. When a record in an entry-sequenced

file is deleted, the block space it occupied is retained and cannot be used for other records. Thus, deleting a record and then reinserting it holds the position of the original record and adds a new record to the end of the file. As a result, deleting and reinserting all records in a file doubles the size of an entry-sequenced file. With a large entry-sequenced file, this may be unacceptable.

If an installation does not want to use a delete-reinsert program for entry-sequenced files, it can combine approaches. It can use the present method to resynchronize key-sequenced and relative files; it can resynchronize entry-sequenced files with the copy program described earlier.

Each deletion and its corresponding reinsertion must take place within the same TMF transaction so that there is no risk of losing data. To make sure it does not manipulate uncommitted data, the delete-reinsert program must use Stable Access to carry out operations on NonStop SQL tables; it cannot use the read-through-lock option on Enscribe files.

Copying many large files or an entire database with RDF can be a lengthy procedure. If possible, the delete-reinsert programs should run when there is minimum demand on the database. This will reduce contention between the database application and delete-reinsert programs. As in the case of online resynchronization with user-developed copy programs, this can also

- Shorten the time required for the resynchronization.
- Improve system response times.
- Increase the bandwidth available for resynchronization over the Expand network.

RDF transmits every delete carried out by the delete-reinsert program to the previously empty database. In each case, the record will not be present for deletion, and the RDF updaters will report a missing-record error (700 File System Error 11). The large number of such messages can make it harder for operators to spot messages for other, more meaningful events.



---

## Resynchronizing While the Database Application Is Stopped

If the database application does not have to run continuously, resynchronization is straightforward. First, one shuts down the application and makes a copy of individual files or the entire database, using one of several methods described below. From this point on, the basic steps are the same as for resynchronization with a user-developed copy program. The application is restarted and RDF is started up in Update Off mode. While RDF transmits updates from the active database to image files on the remote system, the copied data is installed on the remote system. When installation is complete, RDF is switched to Update On mode, and the RDF updaters apply the audit data in the image files to the remote database. As with resynchronization through a user-developed copy program, the copied material must be installed on the remote system before the image disk fills up.

### Using BACKUP and RESTORE

To resynchronize databases using the BACKUP and RESTORE utilities, stop the database application, back up the entire database or selected files to tape, move the backup tapes to the remote site, and restore the remote database. Use either file mode or volume mode to back up an Enscribe database; use only file mode with a NonStop SQL database. Do not use volume-mode backup with a NonStop SQL database. The system and volume name maintained in file labels and the NonStop SQL catalog for the active database are not the same as the names on the backup database. If a volume-mode backup of the active database is restored on the backup database, the path names for database objects will be incorrect.

Before backing up the database to tape and moving the tape to the remote site, one should make a copy of the database on disk. Although this extends the time necessary for resynchronization, it makes tape handling less critical. If a tape error is discovered during the restore operation, one can retrieve the data from disk without stopping the application. Without a copy on disk, one must stop the application and make another backup.

### Using Expand

Users can send a copy of the database to the backup system over an Expand network. This option is particularly useful with files that are relatively small compared to the Expand bandwidth available for communicating with the backup site. One only needs to stop the application briefly and use the FUP DUP command to send small files across the network. A major advantage of this option is the absence of tape handling.

It is often useful to combine this approach with others. For example, one can send a short transaction log directly over the Expand network with the application stopped; user-developed copy programs can transmit large files while the application is running, as described previously.

### **Using Backup Disks**

The fastest way to copy individual disks containing Enscribe files or an entire Enscribe database is to remove the backup disks from mirrored pairs on the intact system and physically move them to the remote system. To avoid running unmirrored, immediately after removing a disk from a mirrored pair, replace it with a substitute disk and revive the new disk with the contents of the remaining disk. Physically moving a disk from one system to another is most feasible with the Tandem 4500 disk subsystem and 4240 and 4230 internal disk storage modules for the Tandem CLX™, Cyclone/R™, and CLX/R™ systems.

The approach described here does not apply to NonStop SQL databases, for the same reason that a volume-mode backup of the primary database cannot be restored on the remote system. In both cases, system and volume names on the primary system will not match the names on the remote system.

---

## **Combining Resynchronization Methods**

In some cases, the most effective approach to resynchronization combines copying part of the database while the application is stopped and part while the application is running. Offline resynchronization provides the fastest way of copying critical data, such as current balance information and lists of stolen credit cards or delinquent customers, from the primary database to the remote database. Less critical information, such as a large transaction log, can be transmitted more slowly, while the database application is operating.

Another approach to combining online and offline methods is to resynchronize the databases volume by volume or by groupings of volumes. In this approach, the database application can stay online throughout the resynchronization. Volumes currently in the process of being resynchronized must be limited to read-only access. Files on all other volumes can be updated. The following example illustrates one way of carrying out volume-by-volume resynchronization. In the example, \$VOL1 on the primary system needs to be copied onto \$VOLA on the backup system. Throughout the procedure, RDF can be in Update On mode.

1. Stop updates to files on \$VOL1, allowing read-only access.
2. Stop RDF after making sure that the RDF extractor has processed all audit data for \$VOL1.
3. Using RDFCOM, remove \$VOL1 from the RDF configuration.
4. Restart RDF.
5. Copy data from \$VOL1 to \$VOLA on the backup system.
6. Stop RDF and use RDFCOM to add \$VOL1 back to the RDF configuration.
7. Restart RDF.
8. Allow updates to files on \$VOL1.

---

## Conclusion

RDF replicates the contents of a primary database on a backup database. This protects data and makes it possible for the backup system to take over operations if the primary system fails. Under certain conditions, the databases can become desynchronized. Such conditions occur infrequently, and in most cases can be prevented through advance measures. When the databases do become desynchronized, there are a number of methods for resynchronizing them. Online methods, which require user-developed software, allow the database application to continue operating while the databases are resynchronized. Offline methods, which are generally faster and more reliable, require a temporary stoppage of the application. In some cases, one can combine both types of methods to optimize the availability of critical data and the duration of the resynchronization.

## References

Guerrero, Jorge. RDF: An Overview. *Tandem Systems Review*. Vol. 7, No. 2. Tandem Computers Incorporated. Part no. 65248.

"Supplemental Information for Release C30." Appendix I, *Remote Duplicate Database Facility (RDF) System Management Manual*. Third edition, 1990; Update 1, 1991. Tandem Computers Incorporated. Part nos. 40595 (manual, third edition) and 62975 (Update 1, includes Appendix I).

---

**Wouter Senf** joined Tandem in 1984 and has since worked on application and database design and performance. He now works for the High Performance Research Center (Frankfurt) and concentrates on availability and contingency planning.

**Frans Jongma** is a database consultant in Tandem's Dutch office. He joined Tandem in 1989. Prior to this, he worked for another hardware company for over eight years as a software developer for database management systems. Frans recently provided technical support in the development of mechanisms for online resynchronization at a major user implementation of RDF.

## Terminal Selection

---

**S**electing the correct terminals for an online transaction processing (OLTP) application can directly affect business objectives. Terminals well-suited for the applications they support and for the system and network environments in which they reside deliver useful function, high reliability, low response time, and the flexibility needed for potential growth. Poorly-suited terminals increase the cost of doing business by wasting the users' time, consuming CPU and network resources unnecessarily, and inflating maintenance and problem resolution expenses.

Terminal selection depends on the type of application using the terminal and on the efficiency and manageability of the operating environment in which the terminal is installed. A terminal suitable for one application type may be unsuitable for a different one. A terminal that works well in one operating environment may not work well in another, and may place too great a load on system resources.

This article discusses terminal selection based on capability and usage. It is intended to assist users inexperienced with data communications in selecting the proper terminal type for specific applications and environments. The article summarizes basic terminal types, discusses criteria for evaluating terminals, and describes each terminal type in detail, weighing the advantages and disadvantages of each alternative.

This is the first of two articles in this issue of the *Tandem Systems Review* that deal with terminal selection and connection alternatives. The companion article, "Connecting Terminals and Workstations to Guardian 90 Systems," provides the data communications specialist with more detailed information on terminals and their connection methods.

## Introduction to Terminal Types and Emulators

Tandem™ supports a broad set of terminals and terminal *emulators*. An emulator is a software package that presents the user and remote host with a representation of a specific terminal's screen, keyboard, and behavior. Emulators run on workstations, such as an IBM PC, a Macintosh, or a UNIX workstation (for example, a SUN). Frequently, the user can run several emulators simultaneously, giving the appearance of more than one terminal on a single screen. Terminals and emulators can be categorized as follows:

- Character-based.
- Page-based.
- Graphics-based.

Character-based terminals are the least adaptable to most OLTP applications. However, they are also the simplest, and usually the easiest, terminals to configure and manage. They transmit characters individually, without blocking them into groups. As a result, character-based terminals frequently require large amounts of data communications and host processing resources.

Page-based terminals are the most suitable for high-volume OLTP. They combine efficient use of the data communications facilities and host processor with an efficient *user interface*, which is the part of a terminal that displays information to and accepts input from the user. Data traveling between terminal and host is transmitted in blocks that usually include data compression and sophisticated error control. Some page-based terminals, and most of their emulators, allow multiple independent screen images to appear on the same screen at the same time, a feature called *windowing*.

Graphics-based terminals are more sophisticated than their character-based or page-based counterparts. However, their use must be carefully monitored to avoid major host performance problems and possible user interface inefficiencies. They are capable of very sophisticated graphics displays under control of the host, and their windowing capabilities may be more advanced than those of page-based terminals or emulators.

## Terminals and OLTP Architectures

Tandem offers two basic OLTP architectures. Both are based on what the industry now calls the *client-server* model, in which interaction with the terminal and interaction with the database are handled by separate processes. The two OLTP architectures are also based on the Pathway transaction processing system, which has, since its introduction, used client-server processing on Guardian™ 90 operating systems.

The first architecture uses Guardian 90 systems for both the terminal-handling and the database-handling processes. (See Figure A.) In this architecture, users interact with the system through terminals connected to the terminal-handling process.

The second architecture was introduced into the Tandem environment in 1985 with the Intelligent Device Support feature of Pathway. In this architecture, processors (such as PCs) that do not use the Guardian 90 operating system perform the presentation-handling function. (See Figure B.)

Recently, the industry has leaned toward the second model, which Tandem supports with a set of tools and interfaces (such as Remote Server Call). However, the majority of current applications, and many that are being designed, use the first architecture, for which correct terminal choice is an important decision.

Figure A

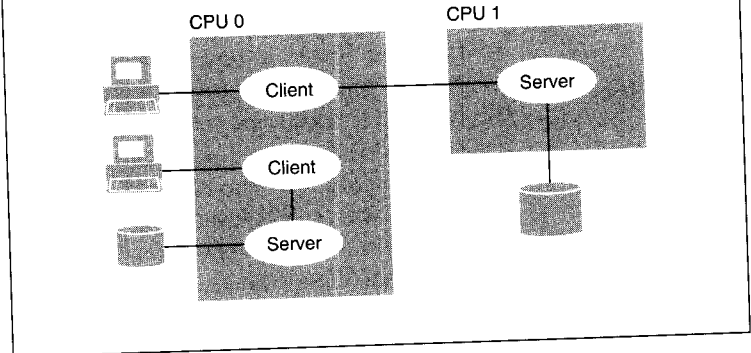
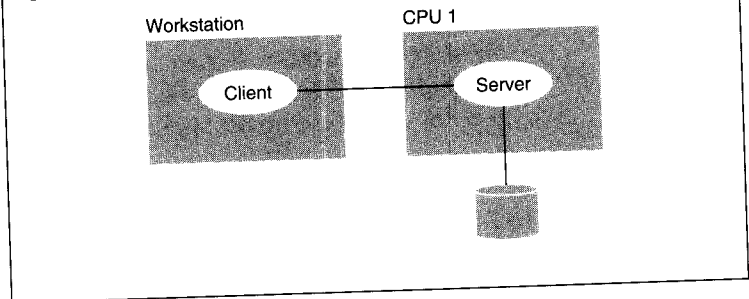


Figure B



**Table 1.**  
Terminal evaluation summary.

Terminal type	Example	Application type				System efficiency and manageability		
		Operations and programming	Decision support	Simple OLTP	Complex OLTP	Host efficiency	Comm efficiency	Comm complexity
Character-based	Teletype	Fair	Poor	Fair	Poor	Poor	Poor	Excellent
Formatted character-based	DEC VT-100	Fair	Fair	Fair	Fair	Very poor	Poor	Very good
Page-based	6530	Good	Good	Excellent	Excellent	Excellent	Excellent	Very good
Windowed	6530 emulator	Very good	Very good	Very good	Very good	Excellent	Excellent	Very good
Graphics	X-window	Excellent	Excellent	Fair	Fair	Very poor*	Very poor*	Fair

\*Assumes native-mode graphics, not terminal emulation.

## Terminal Evaluation Criteria

Terminals should be evaluated by two types of criteria, based on how the terminals are used and how they affect the system in which they are installed. Table 1 shows the relative effectiveness of the terminal types according to these criteria. Although any specific terminal can be used for multiple application types, most usages fall into one of the four categories shown in Table 1. System efficiency and manageability criteria (also shown in Table 1) measure how the terminal interacts with the host and the network.

### Application Type

Suitability of a terminal depends on the types of activities users perform, which vary according to the application. For example, a terminal suitable for OLTP applications may be unsuitable for decision support applications.

**Operations and Programming.** The primary activities performed by operators and programmers are file editing, operations logging and monitoring, and some basic command-line interactions. Older systems are line-oriented; newer systems are page-oriented, windowed, or graphics-oriented.

**Decision Support.** Displaying, selecting, and evaluating data to make decisions requires sophisticated, multiwindow presentation, with possible data transfer among windows. Individual interactions in a sequence are likely to be dissimilar, using multiple screens in various combinations.

**Simple OLTP.** This function consists of simple, highly repetitive form-filling operations. Users perform these operations at high speed, with a minimum of keystrokes and mouse movements.

**Complex OLTP.** Like simple OLTP, this function consists of repetitive form-filling operations. Depending on the transaction, it may use many different forms in various sequences. Important factors are high speed, minimum keystrokes and mouse movements, and quick navigation among forms.

### **System Efficiency and Manageability**

System efficiency measures the load that the terminal places on the host and the network. Manageability measures the load that the terminal places on the person doing network administration. These criteria can be as important as the usage criteria. For example, the designer of a system to be used by large numbers of clerical personnel doing simple OLTP probably would consider system efficiency and manageability to be as important as the application criteria. However, the designer of a system to be used by small numbers of professional staff doing complex decision support might give higher priority to application criteria.

**Efficient Use of the Host.** Most host processor CPU cycles are expensive; therefore, the host should be limited to complex I/O operations and database management and manipulation. In addition, terminal-handling work performed by the host can be resource-intensive and can generate a large number of expensive host CPU interrupts. In some cases, the additional features offered by using host CPU cycles for the user interface may be worth the additional host cost, especially if user efficiency is greatly increased by the host's intensive interaction with the user.

### **Efficient Use of Communications Facilities.**

Two factors that determine communications efficiency in an OLTP environment are the performance of the communications system and the workload that the user interface imposes on that system. Some user interfaces are very sensitive to the communications system. For example, a character-based terminal may use significant amounts of CPU and network resources for each character transmitted. Other terminals have low communications overheads on any communications system. For example, the Tandem 6530 terminal in page-based mode decreases data flow both to and from the host on any communications system. It accomplishes this by storing up to 16 pages of screen formats and data and by transmitting only modified information to the host.

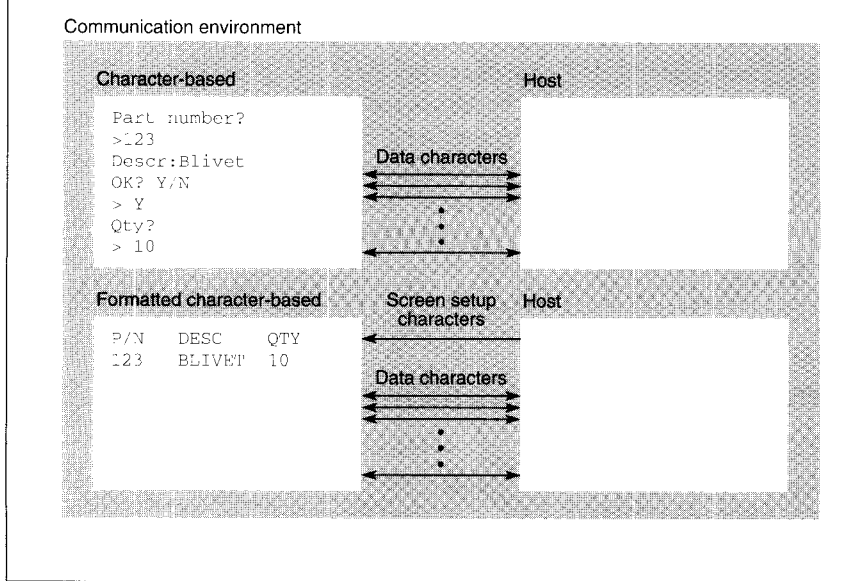
**Complexity of Communications Facilities.** In most cases, terminals with complex communications facilities may be vulnerable to configuration difficulties and problems that may be difficult to diagnose and resolve. One such example is a large, complex local area network (LAN). However, terminal systems with simple communications facilities may sacrifice features, such as error correction, that are important in some environments.

---

## **Terminals and Terminal Emulators**

In addition to the three basic types of terminals (character-based, page-based, and graphics-based), there are two other types: formatted character-based terminals and windowed terminals. As enhanced versions of the basic types, these two terminal types offer additional features to end users. Formatted character-based terminals, a subset of the character-based group, provide a formatted screen interface. Windowed terminals, a subset of the page-based group, provide multiple end-user sessions with the host.

Figure 1



**Figure 1.**  
*Character-based and formatted character-based terminals send each character to the host as it is typed.*

### Character-Based Terminals

Character-based terminals are generally used only for programming, computer operations, and very simple OLTP, involving only a few data items per transaction. They are not a good choice in decision support or any OLTP beyond the simplest transaction types. Also, they may have an unacceptable error rate. Depending on the design of their associated application process, character-based terminals may be very inefficient in their use of communications and host resources. Unfortunately, this terminal type is sometimes the only available choice when a single terminal must connect to different types of mainframe hosts.

Character-based terminals send each character to the host as the character is typed. Historically the earliest type, these very simple terminals are emulations of typewriter-based terminals. Each line is independent and cannot include specialized data fields; therefore, a formatted page cannot be created on the user's terminal. There are no formatting commands other than the rudimentary formatting characters in the character set. Programmers normally use only the simplest of those formatting characters, such as carriage return (CR), line feed (LF), and form feed.

In the example shown in Figure 1, the following characters are exchanged between the host and the character-based terminal. The characters are sent one at a time, and the host interprets the message from the terminal before responding.

```
Host: "Part number? CR LF >"
Terminal: "1 2 3 CR"
Host: "LF Descr:Blivet CR LF OK? Y/N CR LF >"
Terminal: "Y CR"
Host: "LF Qty? CR LF >"
Terminal: "1 0 CR"
```

The main weakness of character-based terminals is that each operator keystroke is sent immediately, and individually, to the host computer. Some application processes use *character-mode* operation and react to each character as it is typed. An example is an application that expands each typed command to its full length as soon as enough characters have been typed to make that command unique. Because they require special host processing for each character received, such designs can be very inefficient in their use of the host CPU. When connected to a LAN or a wide area network (WAN), character-mode designs impose a high communications cost, because the network must transmit each character in a separate packet. The noticeable transit time over a WAN can irritate users while they wait for host responses to individual letters.



Instead of processing each character separately, the host can be set to do the processing only after it receives an end-of-line carriage return, as shown in Figure 1. The application then receives the entire line for processing. Called *line-mode* operation, this communications technique is common for character-based terminals in most engineering, computer operations, and simple business applications. Line-mode operation uses host and communications resources more efficiently than character-mode operation.

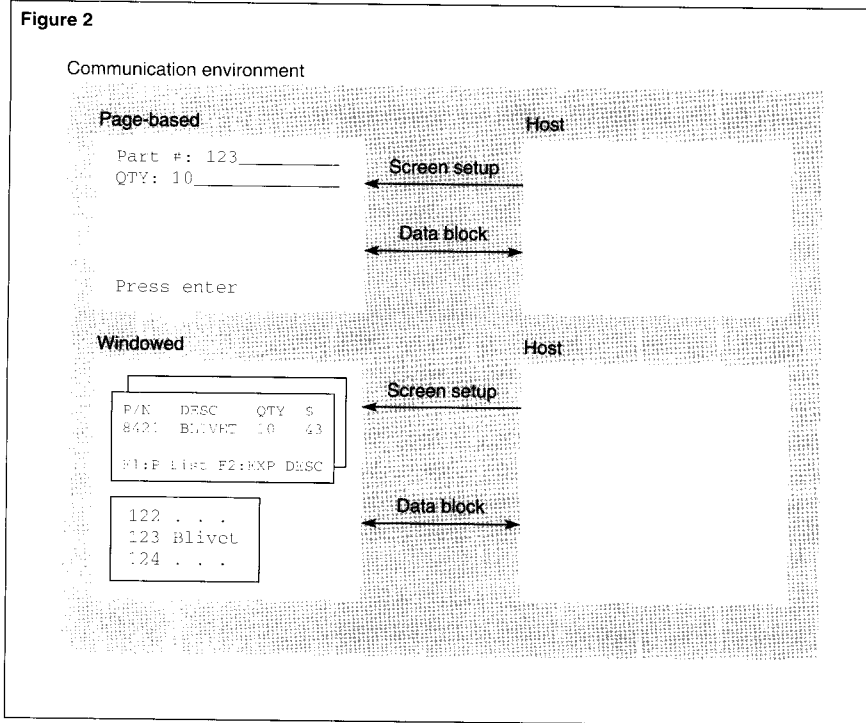
The communications formats and transmission rules (*protocols*) used by character-based terminals usually have rudimentary error detection and lack automatic error correction. Recent attempts to use transmission devices (such as *smart modems*) with error detection and correction protocols do not completely solve these problems. The link between the transmission device and the terminal or host is still vulnerable to error, and the transmission device may be difficult to configure. In addition, the protocols used by most character-based terminals do not handle terminal addressing; therefore, a single communications path cannot be shared by multiple terminals without additional equipment.

Most terminal emulators can emulate character-based terminals. Because it is the only type easily connected to all existing mainframes, virtually all LAN and WAN software supports emulators for this terminal type. Examples are the X.3 emulator for WANs using the X.25 communications protocol standard, and the Telnet Network Virtual Terminal emulator for WANs and LANs using the Transmission Control Protocol/Internet Protocol (TCP/IP) standard.

### **Formatted Character-Based Terminals**

Formatted character-based terminals, such as the DEC VT series and the American National Standards Institute (ANSI) 3.64 (the standard for most simple UNIX terminals), add screen formatting capabilities to the rudimentary character-based terminal. Special character sequences instruct the terminal to move the cursor to specific rows and columns on the terminal screen and to build specialized data-entry fields. The terminal is no longer restricted to the simple emulation of a typewriter-based terminal.

Although these terminals can use their formatting commands to build a full-screen data-entry form for the operator, they are usually an inefficient selection for page-type data entry. (See Figure 1.) Such terminals are appropriate only for small systems with few users, a light workload, and the need to involve the host application program closely with each operator keystroke.



**Figure 2.**  
Page-based and windowed terminals send an entire block of data to the host when the user presses a transmit key.

Most emulators can emulate ANSI 3.64 or DEC VT terminals. Tandem's 6530-type terminals can operate in ANSI 3.64 mode or in a proprietary character-based mode (called *conversational mode*) in addition to their normal page-based operation. The X.25 X.3 emulator and the TCP/IP Telnet Network Virtual Terminal cannot emulate these terminals.

Formatted character-based terminals have the same advantages and limitations as simple character-based terminals, as well as an additional disadvantage. Host efficiency in OLTP can be degraded significantly if the application uses the cursor positioning and other formatting commands to create an entry form on the terminal screen. When a user moves the cursor among fields on the screen to enter or correct data, the host software must handle each cursor movement, which is usually an inefficient use of host resources.

In the example shown earlier in Figure 1, the following characters are exchanged between the host and the formatted character-based terminal. The characters are sent one at a time, and the host interprets the message from the terminal before responding.

Host: "P/N DESC QTY CR LF  
tab\_set tab\_set tab\_set CR"

Terminal: "1 2 3 tab"

Host: "BLIVET tab"

Terminal: "1 0 tab"

Similar to the character-based terminal, the formatted character-based terminal uses communications facilities with only moderate to poor efficiency. Efficiency is moderate when the terminal is connected directly to the host and shares no communications facilities with other terminals. Efficiency is extraordinarily poor over a LAN or WAN, especially if line mode is not used.

### Page-Based Terminals

Page-based (also called *block-mode*) terminals are the best choice for high-volume OLTP and were developed specifically to overcome the disadvantages of the character-based terminal. Page-based communications protocols typically are less error-prone than their character-based counterparts, as they provide more error detection and correction features. They also have a much smaller impact on host performance.

These terminals can store one or more pages within the terminal, have sophisticated formatting capabilities, and usually perform some input error-checking. As shown in Figure 2, page-based terminals send an entire block of data to the host when the user sends an enter command. Also, page-based terminals are capable of transmitting to the host only those screen fields that were modified by the user, thereby decreasing line use and increasing throughput and user-perceived communications speed. Usually, their complexities are handled by OLTP terminal monitors such as IBM's Customer Information Control System (CICS) or Tandem's Pathway transaction processing system. These terminal monitors make applications programming relatively simple by concealing the details of the formatting and terminal-handling commands from the programmer.

On a page-based terminal, the host transmits data and formatting commands to the terminal to create the formatted pages on the screen. Each page contains fields that have associated *attribute flags*, which define the characteristics of each field. For example, the *protected* flag prevents the user from changing the field, and the *modified data* flag marks fields modified by the user for later transmission back to the host. These attributes supplement the common display attributes for conversational terminals, such as *blinking* and *half-intensity*.

The user fills in the fields on the screen, moving among fields with the tab key or cursor keys, and editing fields without any host involvement. When all the data has been entered, the user presses a key to transmit the data to the host. The terminal transmission protocol is highly sophisticated, with high accuracy and low host overhead. Data transmission devices such as smart modems and LAN interfaces may cause problems by interfering with the protocol.

For dedicated use with only one screen at a time (for example, for high-volume data entry or OLTP), a terminal is less expensive and simpler to maintain than a workstation running emulation software. More complex situations may involve multiple screens or concurrent access to different hosts. In such cases, the most common solution is an IBM PC running Microsoft Windows and one or more terminal emulators, connected over a LAN or WAN into one or more hosts.

The Tandem 6530 and the IBM 3270 are page-based terminals in common use on the Tandem Guardian 90 system. Other page-based terminals (such as Burroughs) are used on Tandem hosts, but they are not discussed in this article. They are similar to Tandem 6530s and IBM 3270s, and they are supported by third-party translation software.

### ***Tandem 6530 Page/Conversational Terminals.***

The Tandem 6530 is the most appropriate terminal for complex OLTP applications. Its page-based operation, multiple-screen storage under automatic Pathway control, and error-detecting-and-correcting protocol decrease host workload while ensuring integrity. This terminal functions as either a formatted character-based (conversational-mode) or page-based (block-mode) terminal that is automatically switched between modes by Tandem host software.

The terminal uses a standard asynchronous modem. A cluster controller is not required, although one can be used to increase communications efficiency if terminals are grouped in one location. Emulators are available for all common workstations, and connection over dial-in or direct links is straightforward.

The Tandem 6520, 6524, 6530, 6526, 6526A, 6528, and TS530 terminals are all members of this terminal family. The 6526A

***Page-based terminals are the best choice for high-volume OLTP.***

and subsequent terminals have an ANSI 3.64 mode (for example, for the Tandem Integrity S2™ UNIX-based system) in ad-

dition to the standard Tandem conversational and block modes. Because multiple modes are available, these terminals can be used in a wide range of applications.

Conversational mode is normally used only for simple operator commands. In this mode, the Tandem 6530 terminal can scroll through more than 300 lines; perform automatic data flow control and parity checking; and provide many control sequences for function keys, display formatting, and cursor manipulation. The most suitable applications for this mode are simple one-line commands ending in a carriage return, simple uses for logging-type output, and character-based interactions with a host that does not use the Guardian 90 operating system.

In most OLTP applications, the Tandem 6530 terminal operates in block mode as a page-based terminal. This is the most efficient mode for both user and host, because users enter and edit data without host involvement. The user waits for a host response only after pressing a function key, not after entering each group of characters or line of data. Data is transmitted in blocks, and an error-correcting protocol ensures the proper transmission of all data blocks. Current Tandem terminals and emulators allow the block sizes to be varied because the larger blocks are useful in some network situations.

Communications efficiency is excellent because the Tandem 6530 terminal and its variants store from 8 to 16 screens within the terminal or its emulator. The Tandem Pathway transaction processing system automatically maintains the most recently used set of screens within the terminal. Compared to character-based terminals or to single-page terminals such as the IBM 3270, this capability greatly improves response time for displaying a new screen and decreases communications traffic.

Emulators designed by Tandem to emulate the Tandem 6530 terminal are available for a wide range of workstations, including:

- OS/2-based PCs and Windows-based PCs (the Tandem Terminal Emulator, also called TTE).
- DOS-based PCs (the PC6530 terminal emulator program, also called PCT).
- Macintosh (the Universal Workstation for the Macintosh, MacUWS, from Tandem's Ungermann-Bass™ subsidiary).
- UNIX-based workstations (the TN6530 terminal emulator program).
- X-window graphics terminals with a UNIX-based gateway (the x6530 Terminal Emulator).

Other emulators are available from third-party suppliers.

All the emulators are integrated into their environments. For example, the PC6530 emulator can map terminal field attributes to screen colors, and the OS/2 and Windows emulators support Dynamic Data Exchange. In certain cases, a single workstation screen can simultaneously display multiple terminal emulations. The emulators use the workstation's standard communications ports: asynchronous, WAN-based, or LAN-based.

The Tandem 6530 terminal can be switched under program control between conversational and block modes at any time, making terminal LAN and WAN connections complex. For example, a user may temporarily drop out of a block-mode editor to issue a command to a utility program that runs in conversational mode. Conversational-mode and block-mode protocols are completely different. Most LANs and specialized communications devices (such as smart modems or statistical multiplexers, which combine many terminal connections over a single physical line) may have problems with the two types of data flows. Optimized for one type, they may destroy the other or they may superimpose an unneeded error-correcting protocol that interferes with Tandem's. WANs and Ungermann-Bass LANs are specifically designed to accommodate both protocols and the switching between them. A detailed discussion of this situation appears in "Connecting Terminals and Workstations to Guardian 90 Systems," the companion article by Siegel in this issue of the *Tandem Systems Review*.

**IBM 3270 Terminals.** The IBM 3270 is a page-based terminal, and it is therefore very good for complex OLTP. It shares many of the capabilities of the Tandem 6530 terminal. However, it does not support a character-based operation mode or multiscreen storage in the terminal, and it requires complex communications support.

Because the IBM 3270 does not support character-based operation, host software must perform an emulation if it is required by a particular application program (for example, Tandem's FUP file utility program). The software emulates line-mode character-based operation by creating a page of separate, one-line fields and by using a convention (for example, a line of equals signs) to indicate scrolling down the page. This emulation is not an efficient use of host resources. Furthermore, character-mode operation cannot be emulated at all.

An additional problem is the lack of multi-screen storage. Unlike a Tandem 6530 terminal, an IBM 3270 terminal screen must be transmitted from the host whenever a user switches from one screen to another. Under programmer control, the Tandem 6530 stores up to 16 screens in the terminal.

The IBM 3270's complicated communications requirements arise from its need for a

high-speed link between the terminal and its controller (such as an IBM 3174). That link presents a series of challenges to communications network designers. For exam-

ple, dial-in lines are much too slow to handle the IBM 3270's requirements, requiring the use of special-purpose hardware and software at both ends of the line. Therefore, the dial-in operation of an IBM 3270 terminal or emulator can be awkward and expensive.

The IBM 3270 is so extensively used that almost all WANs and LANs have special methods for handling it with relative efficiency. In page-based applications, the IBM 3270

usually uses both communications and host resources more efficiently than character-based terminals. The IBM 3270's sophisticated protocols ensure that error detection and correction is not a problem.

Tandem's AM3270 access method and SNAX™ (Systems Network Architecture Communications Services) I/O processes handle the IBM 3270 protocols over direct connections and over WANs. Tandem also offers the Guardian TN3270 Server (TN32SERV) to handle IBM 3270 emulators that use the public domain TN3270 protocol over a TCP/IP LAN or WAN. Tandem's Pathway transaction processing system, electronic mail facility, and page-based text editor directly handle the IBM 3270 datastream. Automatic datastream conversion software is built into Tandem's AM3270 and SNAX I/O processes to handle conversion for all Tandem and user-supplied line-mode applications.

### **Windowed Terminals**

Usually, windowed terminals and terminal emulators are best-suited for applications (such as decision support) in which the interaction among host systems does not have a standard pattern and is not used frequently enough to justify building a single user interface. Windowed terminals are also useful for OLTP applications in which multiple terminals must share limited desk space. Generally, windowed terminals are too complex for OLTP. They require too much manipulation, and the windows are too loosely coupled for high-speed, high-volume work.

**Windowed terminals can show multiple independent screen images at the same time.**

Windowed terminals were originally special-purpose hardware that could simultaneously maintain two separate host connections over two separate paths. These terminals are now commonly emulated by workstations running more than one emulator or by an integrated, multiwindow emulator. Each window on the workstation screen presents a separate session. (See Figure 2.) The multiple sessions can connect to the same host or to different hosts.

Windowing depends on the ability of the communications system to provide multiple independent data paths between the workstation and the host. The IBM cluster controller can maintain five separate sessions over a single coaxial cable, allowing most IBM 3270 terminal emulators to provide multiple windows. A single LAN or WAN network connection also can provide multiple sessions. Multiple simultaneous sessions ordinarily cannot be provided over a single asynchronous connection such as a dial-up line.

The multiple windows are managed by the terminal emulator (the usual case for multiple sessions of the same terminal type) or by the workstation's operating system. Some operating system environments provide data exchange facilities among windows and among processes running in the operating system. Many, but not all, of the emulators use these facilities to cut and paste data from one window to another or to a workstation application program. Tandem provides these data exchange facilities in its Macintosh, OS/2, Microsoft Windows, UNIX, and X-window terminal emulators.

Tandem provides two host-based windowing facilities that allow limited windowing over asynchronous lines with standard Tandem 6530 terminals or terminal emulators. The SeeView™ program allows concurrent multiple conversational sessions, and the EM3270 access method combines 6530-to-3270 data translation with multiple windows into IBM and Tandem hosts. SeeView software, which displays multiple conversational sessions simultaneously, can display only one block-mode session at a time. EM3270, which displays only one window at a time, cannot handle conversational mode.

Figure 3

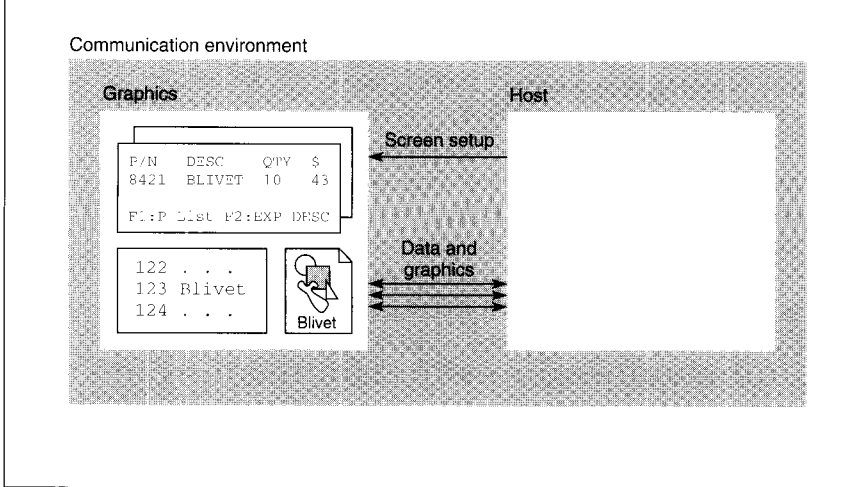


Figure 3.

A graphics terminal can accept commands from an application program on the host to create shapes, typefaces, windows, and other items on the display screen.

### Graphics-Based Terminals

As shown in Figure 3, a graphics terminal, such as an X-terminal, can accept commands from an application program on the host to create various shapes, typefaces, windows, and other items on its screen. It serves the application program and is therefore called a server, reversing the common perception of the terminal as a requester of services. These terminals do have some advantages, but they also have significant disadvantages for non-graphic OLTP situations.

The most efficient use of graphics terminals is to perform windowing in a decision-support or computer operations application. In OLTP, most data-entry personnel quickly tire of the accurate mouse movements required by these terminals. In addition, intensive interaction by the host process with the graphics terminal in *native mode* (not emulating a page-based or character-based terminal) can flood the host with terminal-handling work.

Thus, graphics terminals use more host resources than formatted character-based terminals. Native-mode graphics interfaces are impressive when one user interacts with the host, but they can make major demands on the host as the number of users increases. The response time becomes longer for each added terminal as the host processor must simultaneously handle display details for the increasing number of terminals.

Tandem provides special emulation software (the x6530 Terminal Emulator) for the X-window graphics terminal standard. This emulator runs on any standard UNIX system as an intermediary between the X-window terminal and the Tandem Guardian 90 host. The Guardian 90 host communicates using standard TN6530 block-mode and conversational-mode protocols over Ethernet to the x6530 software. The x6530 software converts the data into X-window commands over Ethernet and sends them to the X-window terminal. The flood of interruptions from the X-window terminal is handled by the UNIX system, not the Tandem Guardian 90 system.

Instead of buying an X-terminal, many users choose to run X-window emulators on a UNIX workstation. In such cases, the X-window emulator and the x6530 software can run in the same workstation.



## Conclusion

Properly chosen terminals can enhance the productivity of both the user and the host computer system. In most cases, the Tandem 6530 terminal or one of its emulators is the best choice for high-volume OLTP. The IBM 3270 terminal is a second-best choice.

However, other terminal types (character-based, windowed, or graphics) are indicated in some special situations. Character-based terminals and emulators are effective for simple, low-volume interactions with the host. Windowed terminals perform most efficiently for complex decision-support or programming environments. Graphics-based terminals offer the best solution for integration into workstation architectures dependent on these device types for other applications, such as image-retrieval systems.

For dedicated use with only one screen at a time (for example, high-volume data entry or OLTP), a terminal is less expensive and simpler to maintain than a workstation running emulation software. More complex situations may involve multiple screens or concurrent access to different hosts.

## Acknowledgments

The author would like to thank Bruce Burns, Doug Humphrey, and Janet Smith for their work reviewing the article.

---

**Eric D. Siegel** joined Tandem in 1984 and is currently with the Integration and Technology Center, where he provides consulting services to Tandem customers. He wrote the core data communications course for Tandem analysts and is the corporate focal point for all data communications specialists. Eric has BS and MEng degrees in Electrical Engineering from Cornell University.

## Connecting Terminals and Workstations to Guardian 90 Systems

**T**he Tandem™ Guardian™ 90 operating system supports many alternatives for connecting terminals, terminal emulators, workstations, and other user interfaces to the Tandem host. The number of connection alternatives has grown as Tandem users request support of nonproprietary standards and of proprietary methods from IBM and other vendors.

The many available alternatives provide the flexibility to find an optimal solution to almost any set of requirements. This article provides Tandem users with the information needed to navigate through those alternatives and determine the best design for their requirements.

This article provides an overview of the connection options for terminals and terminal emulators running on workstations. This is the second of two articles in this issue of the *Tandem Systems Review* that deal with terminal selection and connection alternatives. It assumes familiarity with the companion article, "Terminal Selection," and with the concepts and facilities of telecommunications systems. The companion article is intended to assist users in selecting the proper terminal type or types for specific applications and environments.

---

## Evaluation Criteria for Connection Designs

Users can choose from many different designs to connect a terminal or terminal emulator to a Tandem Guardian 90 system. Depending on their requirements, users need to evaluate their connection designs according to certain criteria: the geographic distribution of users; concurrent session support; communications complexity, reliability, maintainability, and error rate; interoperability; suitability for the Guardian 90 system; performance and host processor use; and cost of ownership.

### Distributed User Location Support

There are four typical distribution patterns for user locations. Each pattern lends itself to a different set of terminal connection designs:

- Local distribution places the terminals within 1500 feet (500 meters) of the host.
- Campus distribution typically places the terminals in many separate locations, all within approximately 2 miles (3 kilometers) of the host.
- Grouped remote distribution places the terminals more than 1500 feet (500 meters) from the host in groups that can share common communications equipment.
- Isolated remote distribution places the terminals more than 1500 feet (500 meters) from the host, but not close enough to each other to share common communications equipment.

### Concurrent Session Support

Many users require the ability to run concurrent multiple terminal sessions on a single workstation. This facility, called *windowing*, normally requires multiple independent communications paths, or *sessions*, between the workstation and the host.

### Complexity, Reliability, Maintainability, and Error Rate

A high degree of communications complexity usually implies lower reliability and more difficult maintenance. Some widely used connection designs are exceptions to this general rule, because their behavior is well understood and specialized operations tools are available.

A criterion of reliability is the error rate encountered during network operations. Guardian 90 I/O processes attempt to ensure zero errors in the communication path; however, some connection designs are more likely than others to achieve a zero error rate.

### Interoperability

Interoperability allows the Guardian 90 host to communicate with terminals over an existing communications facility. This capability simplifies system management and decreases user costs.

### Suitability for Guardian 90

The Guardian 90 operating system, designed for efficient and reliable online transaction processing (OLTP), can handle almost any method for connecting terminals. Some methods, however, use specialized equipment that the Tandem system does not need and that may cause communications difficulties. An example is the smart modem, which incorporates automatic techniques to correct transmission errors. Although simpler terminals with no error-correction capability may need these devices, they may only interfere with Tandem terminals, which have built-in error correction.

**Table 1.**  
Terminal connection designs.

Design groups	Designs	Options
<b>Direct local</b>		
	EIA-232	--
	Current loop	--
<b>LAN</b>		
	Terminal server	--
	TCP/IP	--
	NETBIOS	--
<b>Group</b>		
	Multiplexer	TDM, statmux
	Multidrop and cluster controller	AM6520, IBM bisync, 6600 cluster controller, 3270 cluster controller
	X.25 direct (no network)	X.3 PAD, 3270 cluster controller
<b>Direct remote</b>		
	Unswitched	--
	Switched	--
	Multispeed and intelligent modems	--
	Autodial	--
<b>WAN</b>		
	X.25 network	X.3 PAD, TCP/IP Telnet, SNA over X.25
	SNA network	Cross-Domain Facilities

### Performance and Host Processor Use

Some connection designs use fewer Tandem host resources than others doing an equivalent amount of data handling, and some designs provide a faster response time than others. Because

most host CPU cycles are expensive, host terminal-handling, with its many resource-intensive interrupts, should be minimized. The host should perform only complex I/O operations and database management and manipulation.

### Cost

Users should consider the entire system cost, both initial and ongoing. A terminal and network implementation may have a low initial cost. However, a system may be more expensive than it first appears if it fails often, takes a long time to repair, or requires inefficient use of the keyboard or mouse.

## Terminal Connection Designs

Table 1 shows five types of terminal connection designs. The five design types are as follows:

- **Direct local:** a terminal directly connected without any intervening communications equipment (such as modems) and without sharing a communications path.
- **Local area network (LAN):** a terminal connected over a LAN and associated equipment.
- **Group:** a group of local or remotely located terminals connected through communications concentration equipment such as a pair of multiplexers or a cluster controller.
- **Direct remote:** a terminal with a modem directly connected through a dial-up or leased line.
- **Wide area network (WAN):** a remotely located terminal connected through the facilities of a WAN.

Table 2 evaluates the designs with respect to some of the criteria discussed in the preceding section. The table does not summarize all criteria, because of their complexity; however, it provides a good starting point for matching a particular case to candidate designs.

## Direct Local Connections

Virtually all directly connected terminals use asynchronous communications, which do not use a separate interface wire to synchronize the sending and receiving devices. Local asynchronous connections use either the EIA-232 interface or the 20-milliamp current-loop interface. The EIA-232 standard is equivalent to the V.24/V.28 standard of the Consultative Committee for International Telephone and Telegraph (CCITT). Figure 1 shows a direct local connection.

### Evaluation

The direct local connection design provides excellent support for local users but is generally poor for all other location patterns. It does not normally support concurrent sessions; therefore, windowing terminals usually need more than one connection or another solution. Complexity is low, reliability is high, and troubleshooting is simple.

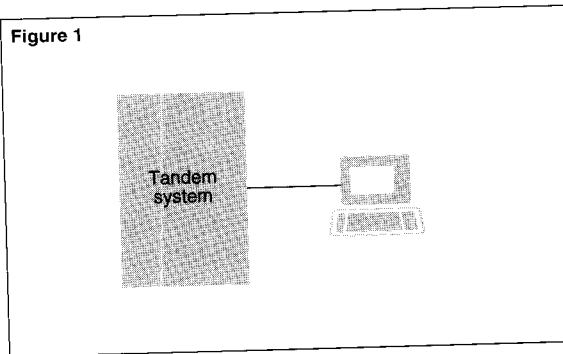
Interoperability with existing communications facilities is excellent; both the EIA-232 and current-loop interfaces are standard and easily carried by all wiring systems. The direct local connection design is excellent for Tandem 6530 terminals and for character-based terminals operating in line mode. Performance is good, and host communications resource requirements are low if character-mode and formatted character-mode operation are not used. The effective error rate is virtually zero for Tandem 6530 terminals operating in block mode because of Tandem's proprietary error-correcting protocol. For 6530 terminals in conversational mode and for character-based terminals, errors can usually be detected, but not automatically eliminated. The cost is low because only wire connections are required.

**Table 2.**  
Design criteria evaluation and summary.

Design group	Evaluation criteria					Concurrent session support	Complexity, reliability, maintainability
	Location support			Grouped remote	Isolated remote		
	Close local	Campus					
Direct local	1	2	-	-	3	1	
LAN	2	1	1	-	1	3	
Group	2	1	1	-	1	2	
Direct remote	3	2	3	1	3	1	
WAN	3	2	2	1	1	2	

1 = Probable candidate  
2 = Acceptable candidate  
3 = Requires careful evaluation

**Figure 1**



**Figure 1.**

*Direct local connection (EIA-232 or current loop).*

### Connection Components

EIA-232 is designed for modem connections, but the modems can be replaced by a specially wired *null modem* cable. The EIA-232 cable's electrical characteristics typically restrict it to a 50-foot (15-meter) distance. Special cables with improved characteristics (for example, shielding and low capacitance) may be able to handle distances up to ten times as great.

The current-loop connection is preferred for local distances of up to 1500 feet (500 meters). It is simple, reliable, and requires only four wires, often sharing telephone wiring facilities.

An inexpensive adapter available from the Tandem Source Company can convert a current-loop connection into an EIA-232 connection at the back of the terminal or workstation. For both the EIA-232 and the current-loop connections, the maximum recommended data transmission speed is 19,200 bits per second.

Tandem 6530 terminals and virtually any other character-based terminal can be handled easily on asynchronous connections. The Guardian 90 I/O process that handles asynchronous terminals for the older Tandem 6303 and 6304 asynchronous controllers is called TERMPROCESS. The equivalent ATP6100 communications access process handles the newer intelligent controllers (such as the Tandem 6106 Asynchronous Communications Controller, 3606 controller, and 3650 Communications Subsystem). XON/XOFF flow control is available in the intelligent controllers, with selectable XON and XOFF characters.

Uncommon cases (such as unusual blocking formats) can be handled by the Envoy™ I/O process, the Generalized Full Duplex protocol available in the CP6100 generalized access method I/O process, or by third-party software. The Envoy I/O process requires a 6202, 6303, 6304, 3602, or 3603 controller. The CP6100 I/O process requires an intelligent controller.

Character-mode terminal operation, which interrupts the Tandem host for each character transferred, is very inefficient and should be avoided. Line-mode operation is far more efficient in its use of Tandem host and communications resources.

Formatted character-based operation (for example, pseudo-page-mode operation of an ANSI 3.64 or DEC VT-100 terminal) requires specialized software. Because of the number of interruptions such terminals create in host processing, this mode also can be quite inefficient in its use of Tandem host resources. Tandem can provide custom software for situations in which a Tandem 6530 terminal's page-based block-mode operation must be simulated.

Standard IBM 3270 terminals cannot operate without an IBM cluster controller, but many 3270 emulators can. Special hardware attached to the Tandem host emulates an IBM cluster controller, and workstations running associated emulation software connect to the controller emulator.

Theoretically, a graphics terminal can operate over an asynchronous connection. However, the low speeds of such transmissions (typically less than 20,000 bits per second) make such a connection impractical.

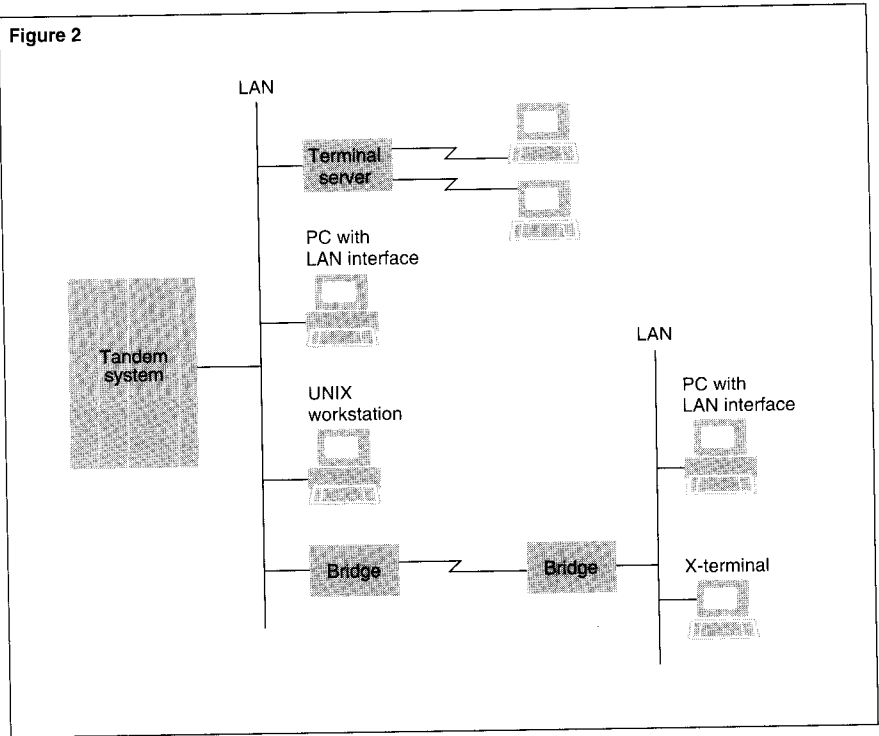
## LAN Connections

In the past few years, LAN connections have become quite common. A LAN is a high-speed network, typically operating at 10 megabits per second and usually contained within a single building or campus. Dissimilar sets of users can share a LAN without interfering with each other. For example, DEC VT terminals can connect to a DEC VAX host on the same LAN used by Tandem 6530 terminals to connect to a Guardian 90 host. Such architectures can be very effective in a multiple-mainframe environment. However, the stations must all use a compatible LAN protocol and have unique LAN addresses. Figure 2 shows a LAN connection.

### Evaluation

Evaluation of LAN-based designs can be complex. LANs provide excellent support for campus and grouped remote situations. LANs can be bridged or routed over long distances, allowing several LANs in different areas to function as if they were one LAN. However, the throughput and response time will vary depending on the load on the LANs and on the bridges or routers. LANs support close local distributions well, but simpler alternatives (such as direct local designs) may be preferable. Support for concurrent sessions is excellent for Transmission Control Protocol/Internet Protocol (TCP/IP)<sup>1</sup> and network basic input-output system (NETBIOS) designs; however, concurrent session support is not normally available for terminal server designs.

Figure 2



Managing and maintaining a large LAN can be a full-time job requiring sophisticated tools and training. Therefore, the installation and operations cost may be higher for LANs than for the alternatives.

Figure 2.  
LAN connection.

<sup>1</sup> TCP/IP is the U.S. Department of Defense suite of protocols.

Interoperability with existing communications facilities is excellent and highly flexible. All the LAN design access methods can co-exist with connections other than Tandem over the same physical connection. For example, MS-DOS workstations through a Multilan™ connection, SUN/4 workstations and X-window graphics terminals using TCP/IP, and Tandem 6530 terminals connected through asynchronous terminal servers can attach into a single 3613 Ethernet/IEEE 802.3 I/O controller on the Tandem host while a DEC terminal communicates with a VAX on the same LAN. A single LAN connection to a workstation allows concurrent, windowed access to different hosts using different terminal emulators simultaneously.

LAN suitability for the Guardian 90 system is excellent for Tandem 6530 and IBM 3270 terminals and for character-based terminals operating in line mode. LAN software performance on the Guardian 90 operating system is very good, but a LAN is more complex and therefore less efficient than an asynchronous, multidrop, X.25, or SNA design.

The effective error rate is virtually zero for IBM 3270 terminals and Tandem 6530 terminals in block mode because of their proprietary error-correcting protocols. For terminal servers handling 6530s in conversational mode or character-based terminals, errors can appear in the link between the terminal and the terminal server. Errors are virtually zero for other designs. LAN cost is moderate, depending on the complexity of the LAN.

### **LAN Protocols**

The most common LAN protocols are Ethernet and IEEE 802.3 for the lower communications layer (which provides basic connectivity) and TCP/IP and Xerox Network System (XNS) protocols for the upper communications layers (which provide user services). Ethernet and IEEE 802.3 can share a single LAN, although an IEEE 802.3 station cannot communicate with an Ethernet station. The TCP/IP protocols handle terminal communications, file transfer, electronic mail, and some other functions. Both TCP/IP and XNS normally use Ethernet as their lower layer.

The XNS and TCP/IP protocols perform similar functions but are incompatible; a station using XNS cannot communicate with a station using TCP/IP. Because of their popularity in the UNIX community, the TCP/IP protocols are replacing the XNS protocols in many networks. Ungermann-Bass™ (U-B) systems can handle both TCP/IP and XNS protocols concurrently in the same LAN and even in the same workstation interface.



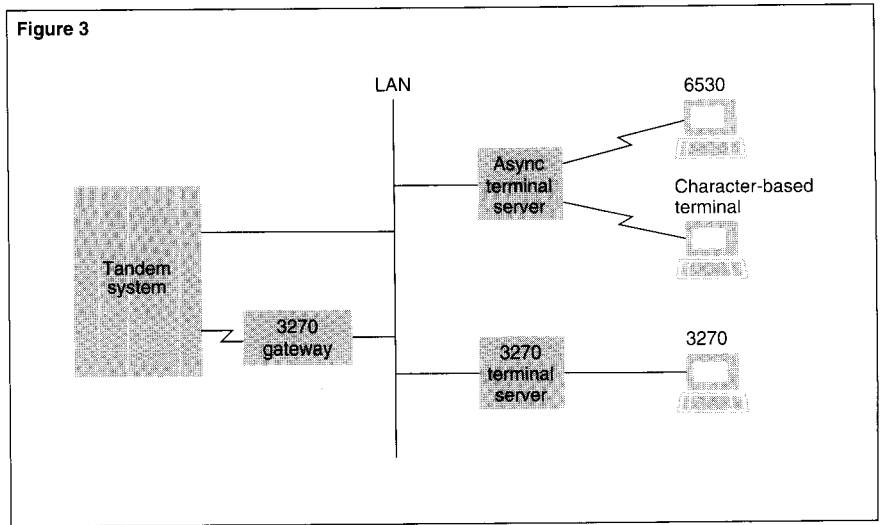
LANs in different areas can be joined by bridges or routers using direct or WAN links, but capacity limits in the bridges or routers and speed limitations on the links may cause throughput problems. Because the direct or WAN link speeds are usually much slower than the LAN speeds, the bridges or routers must be able to manage the flow of traffic to avoid losing data. This disparity does not present a problem at moderate LAN usage levels; however, large file transfers or other heavy loads that exceed device capacity can force the bridges or routers to discard data, which then may be automatically retransmitted by the communicating user devices.

Many different protocols are used within LANs and with LAN connections to a Tandem system. However, the interfaces between a Tandem host and its associated terminals and terminal emulators can be divided into three classes:

- Terminal servers.
- TCP/IP.
- NETBIOS.

### Terminal Servers Through LANs

Terminal servers provide the only method available for connecting terminals and workstations that do not have built-in LAN interfaces. This section discusses terminal servers for asynchronous terminals and for IBM 3270 terminals. Figure 3 is an example of terminal server connections through LANs.



**Asynchronous Terminal Servers.** These servers, such as a U-B NIU™-190 Network Interface Unit or a U-B ASM-100 module within an Access/One™ system, connect asynchronous terminals to LANs. The Tandem host connection also uses an asynchronous terminal server. For a U-B LAN, the Tandem host can use the TCP/IP I/O process and devices such as the 5600 or 3613 Ethernet/IEEE 802.3 controllers.

**Figure 3.**  
*Terminal servers through a U-B LAN.*

Most asynchronous terminal servers for LANs interfere with the error-correcting protocol used on Tandem 6530 terminals and have problems when the 6530 terminal switches between conversational and block modes. The current U-B LANs and asynchronous terminal servers do not have these problems, but older U-B asynchronous terminal servers such as the NIU-130 and NIU-180 Network Interface Units cannot handle Tandem 6530 terminals in block mode.

Character-based terminals, with their virtually nonexistent protocol, typically have no difficulties with asynchronous terminal servers. Performance is best in line-mode operation. Performance in character mode may be poor because of the massive amount of resources required to transmit each character in a separate LAN packet.

Formatted character-based terminals usually need custom software in the Tandem host to simulate a 6530 in block mode, and performance is equivalent to that of character-mode terminal operation. It is preferable to operate these terminals in line mode.

ANSI 3.64 and DEC VT-type terminals using a terminal server are a special case of formatted character-based terminals. If the terminal server uses the DEC-proprietary Local Area Transport (LAT) protocol, a gateway from a Tandem Alliance member can connect that server to the

Tandem host while emulating a Tandem 6530 on the server's terminals. The Tandem Pathway transaction processing system and other similar block-mode applications can then run without change on the terminal. The gateway, not the Tandem host, handles the complex task of emulating a block-mode Tandem 6530 terminal on a formatted character-based terminal.

An alternative for attachment of ANSI 3.64 and DEC VT-type terminals to Tandem is a terminal server that emulates an IBM 3270 on its attached terminals. As IBM 3270s, they are easily handled by Pathway. Two examples are the U-B 3270 Remote Gateway and terminal servers that use the public domain TN3270 protocol over TCP/IP on their network link. The TN3270 protocol is handled by the Tandem Guardian TN3270 Server (TN32SERV) product.

**IBM 3270 terminals.** These terminals can be connected to their IBM cluster controller by using a LAN. However, they require a special adapter at both the terminal and at the IBM cluster controller to handle their coaxial cable connections. Examples are the U-B NIU-78 Network Interface Unit or a U-B ASM-200 module within an Access/One system for the terminal and the NIU-74 Network Interface Unit for the cluster controller. The cluster controller connects to the SNAX™/XF (Systems Network Architecture Communications Services/Extended Facility) I/O process on the Tandem host by using a synchronous data link control (SDLC) line. Guardian 90 applications generally handle the IBM 3270 data stream directly.

To avoid using an IBM cluster controller, a U-B 3270 Remote Gateway directly connects a Tandem SDLC link to a U-B LAN. The NIU-78 Network Interface Unit or a U-B ASM-200 module within an Access/One system provides the coaxial connection to the IBM 3270 terminals. The U-B 3270 Remote Gateway also provides a protocol conversion service. This service allows ANSI 3.64 and DEC VT-type terminals attached by the NIU-190 Network Interface Unit, Access/One's ASM-100 system, or other U-B interfaces to communicate with the Tandem host, and with IBM hosts, in 3270-emulation mode.

## TCP/IP Connections Through LANs

TCP/IP connections use the U.S. Department of Defense TCP/IP protocol suite layered over the Ethernet protocol. Most workstations and some terminals can accept an interface card that allows them to connect directly to a LAN using the TCP/IP protocols. Figure 4 is an example of TCP/IP connections through LANs.

The TCP/IP protocol suite includes the Telnet protocol (designed for terminal connections), which in turn contains a simple character-based terminal emulator, the Network Virtual Terminal (NVT). Tandem hosts can handle the NVT in character mode or line mode, but, because it has no formatting capabilities, the NVT cannot simulate a page-based terminal.

Formatted character-based terminal emulators, such as those for the DEC VT-100, are usually available in TCP/IP protocol packages. These emulators can simulate a page-based Tandem 6530 terminal if the Tandem host uses custom software, but efficiency is very poor because the emulators must transmit each character in a separate LAN packet.

Tandem has built emulators to run on commonly used workstations and emulate the Tandem 6530 terminal directly. These emulators are available for:

- OS/2-based PCs and Windows-based PCs (the Tandem Terminal Emulator, also called TTE).
- DOS-based PCs (the PC6530 terminal emulator program, also called PCT).
- Macintosh (the Universal Workstation for the Macintosh, MacUWS, from Tandem's Ungermann-Bass™ subsidiary).
- UNIX-based workstations (the TN6530 terminal emulator program).
- X-window graphics terminals with a UNIX-based gateway (the x6530 Terminal Emulator).

The workstation side of the interface requires an adapter card or other LAN attachment, TCP/IP protocol software, and the appropriate terminal emulator. The Tandem side of the LAN connection for TCP/IP is the same as for asynchronous LAN connections (a Tandem Ethernet I/O controller and the TCP/IP I/O process).

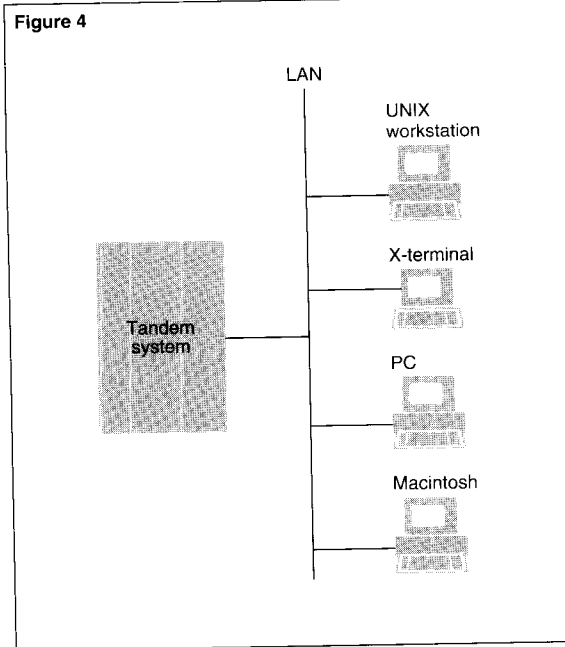


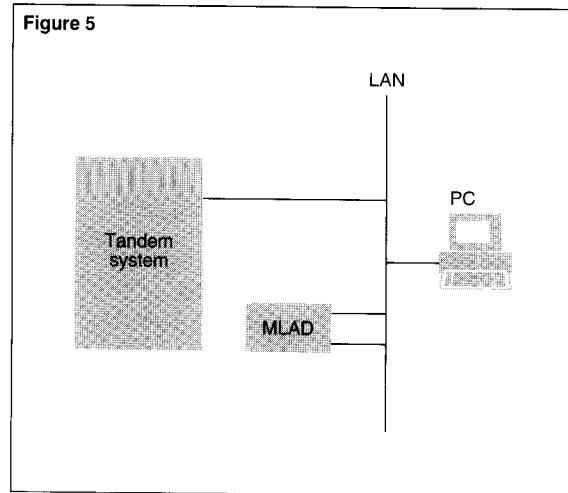
Figure 4.

*TCP/IP connection through LANs using emulators for Tandem 6530, Telnet NVT, and IBM 3270.*

Some workstations and terminal servers are equipped with IBM 3270 terminal emulators that use the public domain TN3270 protocol running over TCP/IP and Ethernet. These emulators can connect to the Tandem host through an Ethernet I/O controller, the TCP/IP I/O process, and Tandem's TN32SERV software. Both block-mode and simulated conversational-mode operation are supported.

Various Tandem Alliance partners build alternatives to Tandem's TCP/IP products. These Alliance products should also be considered for unusual cases such as Tandem 6530 terminal emulation on atypical workstations.

**Figure 5.**  
*NETBIOS connection  
through LANs.*



### **NETBIOS Connections Through LANs**

NETBIOS connections use an interface within PC-based workstations that are at a higher level than the TCP/IP or XNS protocol stacks. Thus, the underlying protocols are not visible to the terminal emulator. Using NETBIOS permits a Tandem terminal emulator to connect to a Guardian 90 host over any network supporting the NETBIOS interface standard, which includes almost all DOS, Windows, and OS/2 workstation networks. Figure 5 is an example of NETBIOS connections through LANs.

The Tandem product that enables this operation is a Multilan connection, which consists of several software programs. One program runs in a dedicated PC, the Multilan attachment device (MLAD) on the LAN. The other program runs in the Tandem host. The dedicated PC is equipped with the LAN hardware and software, and it uses Ethernet to communicate with the Tandem host Ethernet controller. The MLAD software forms a bridge between the LAN and the Tandem host. The Multilan connection has passed rigorous tests in over 30 environments, including Banyan Vines 4.0 over 3Com, Novell over IBM Token Ring, 3+Open LAN Manager over 3Com, and U-B.

Any PC with a NETBIOS interface on the LAN can then use the MLAD to reach a NETBIOS interface inside the Tandem host. This configuration provides a programmatic NETBIOS interface for building cooperating processes on the Tandem host and the PC. The terminal emulator, which can run in one or more windows on the PC, uses the NETBIOS interface. Tandem also supports the Microsoft Server Message Block (SMB) protocol, which allows PC users to access Tandem printers and disks as if they were PC resources. LANs that do not support the standard SMB protocols (such as Novell or Banyan) can support only the NETBIOS interface and the terminal emulator services of the Multilan connection. These LANs cannot support the file and print services unless the standard Microsoft SMB service handlers are loaded into the workstation along with the proprietary handlers.

### **Comparison of LAN Alternatives**

Analysis of LAN suitability is complex and normally depends on more than Tandem connectivity considerations. Tandem hosts connect to most LAN architectures very well.

If terminals or workstations with asynchronous connections must be supported, the only available option is to use asynchronous terminal servers, which do not normally support parallel sessions. If Tandem 6530 terminals must be supported, the appropriate U-B terminal servers are strongly recommended. If Tandem 6530s are not supported (the LAN is restricted to character-based or IBM terminals), then any LAN supporting paired terminal and host servers will work well.

Most LAN connections are not asynchronous and contain workstations with direct LAN connections. Such workstations easily run parallel sessions for multiple windows, and can perform file transfer and other functions easily and concurrently by using terminal emulation. Workstations running NETBIOS, normally restricted to PCs with DOS, Windows, or OS/2, are usually best served by the Multilan connection, which, if the Microsoft SMB protocol is available, provides file and print service along with terminal emulation and process-to-process communications.

Workstations not supporting NETBIOS must use the TCP/IP interface. Typical examples are a Macintosh (running MacUWS) and a UNIX workstation (running the TN6530 terminal emulator program). Some NETBIOS workstations may use the TCP/IP interface if they cannot support full Multilan functionality or if they have architectural reasons for avoiding the use of the MLAD gateway.

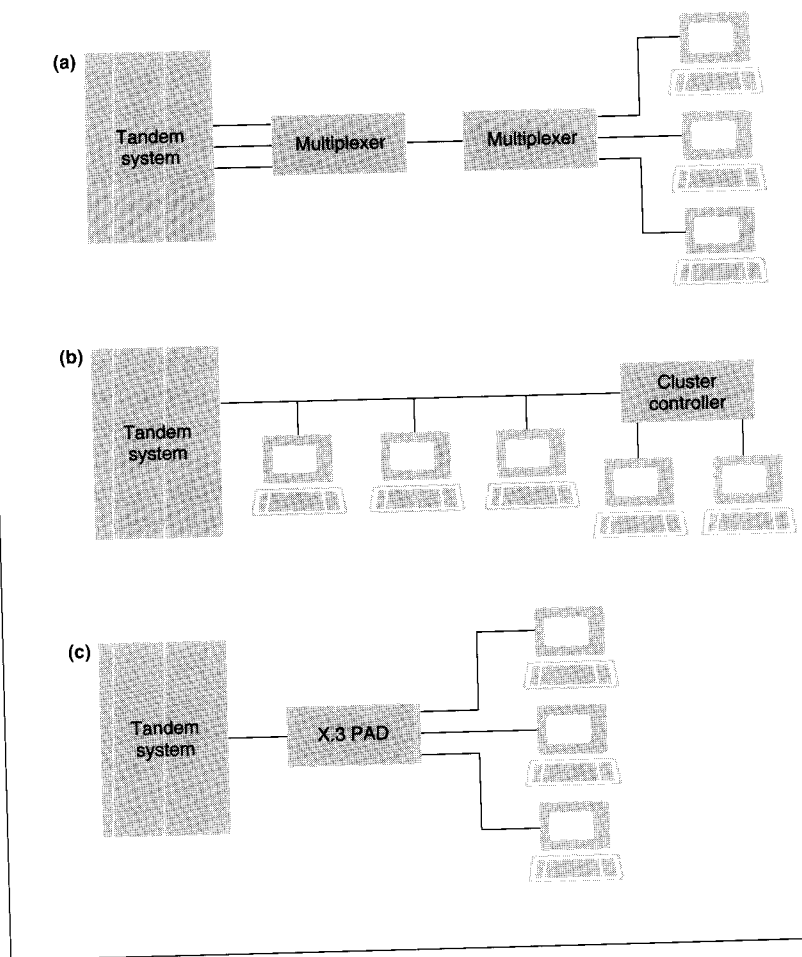
X-window graphics terminals must connect to a UNIX host running the Tandem x6530 Emulator Program, which then uses TCP/IP over Ethernet to connect to the Guardian 90 host. Both the X-window connection to UNIX and the UNIX connection to the Guardian 90 system normally run over the same Ethernet. Sometimes, the X-window terminal is emulated on a UNIX workstation. The user can then choose either the TN6530 terminal emulator program or the x6530 Terminal Emulator in the workstation to handle the workstation's emulated X-window terminal.

## Group Connections

Grouped communications normally are used when multiple terminals are at the same physical location. Many systems are now being built with the cluster controller as the center of a dial-in hub. Often, it is less expensive to make switched local telephone calls to a cluster controller than to have separate dedicated lines to each terminal. As shown in Figure 6, three types of groupings are in common use:

- Multiplexing.
- Multidrop and cluster controllers.
- Direct X.3 packet assembler and disassembler (PAD) connections.

Figure 6



## Evaluation

For these three designs, the support of distributed user locations is excellent for campus and grouped remote situations. Support is poor for isolated remote users because of the cost of the clustering devices unless dial-in to clustering devices is possible. Although this option supports close local situations well, simpler alternatives, such as direct local designs, may be preferable.

Figure 6.

Group connections. (a) Multiplexer connection. (b) Multidrop and cluster controller connection. (c) X.25 direct connection.

Support for concurrent sessions is excellent for X.25 PAD emulators and for cluster controller emulators embedded in the workstation; the workstation can emulate multiple terminals concurrently. This support is not usually available for multiplexer designs or for any other design using asynchronous connections between the terminal and the clustering device. IBM 3270 terminals and their emulators often have a multiple window capability that can support concurrent sessions over the coaxial cable between the terminal or terminal emulator and a PAD or cluster controller. The cluster controller and X.25 PAD alternatives are stable, and their configuration and management are straightforward. Statistical multiplexers (statmuxes) present more problems, and debugging them may be more complex.

IBM 3270 cluster controllers and Tandem 6600 Intelligent Cluster Controllers (for 6530 terminals) can share the same physical link, providing interoperability with existing communications facilities. The link can be multiplexed over properly configured time division multiplexers (TDMs) and statmuxes. The X.25 PAD design can also be handled by any multiplexer that can be configured to handle the X.25 link protocol. The suitability of statmuxes for Tandem is mixed, but the suitability of cluster controllers and X.25 is excellent. Cluster controllers handle Tandem and IBM 3270 terminals. X.25 PAD direct connections support character-based, Tandem 6530, and IBM 3270 terminals. Tandem's Pathway system, electronic mail facility, and page-mode text editor all handle the IBM 3270 data stream directly. Automatic data stream conversion software handles 3270 conversion for all Tandem and user-supplied conversational-mode applications.

Host protocol-handling overhead is relatively low, regardless of the clustering method and I/O controller used. All Tandem controllers handle the AM3270 and AM6520 Access Methods and their polling in the controller. The only appreciable host overhead occurs when an application is using the line-mode data stream conversion facilities for IBM 3270 terminals (an uncommon situation).

The effective error rate is virtually zero for IBM 3270 terminals and Tandem 6530 terminals in block mode because of their proprietary error-correcting protocols. The error-handling characteristics of character-based terminals may be enhanced by using statmuxes or X.25 designs, although the additional buffering will increase response time slightly. Errors can still appear in the link between the terminal and the clustering device for all character-based terminals. Clustering devices are an additional expense, but the savings in physical line costs easily justifies their use for groups of terminals.

## Multiplexer Connections

Multiplexers combine multiple data links on one physical line, but do not decrease the number of host ports needed. This connection requires a pair of multiplexers, one at each end of the link. (See Figure 6.)

Often, a distributed Tandem CLX™ system using Tandem's fully distributed NonStop™ SQL relational database management system is less expensive and more resilient than a network of multiplexers. Two multiplexer types in common use are time-division multiplexers and statistical TDMs.

**Time-Division Multiplexer (TDM).** TDMs are completely transparent to the terminal and to the host. They add virtually no delay to transmission, and their complete transparency allows different protocols to be mixed on the same physical link without any interference. The only drawback to a TDM is that its capacity allocations among the data streams is fixed. A 2400-bits-per-second allocation does not change to 4800 bits per second without TDM reconfiguration, and the total capacity allocations cannot exceed the total capacity of the physical link. TDMs are inexpensive, and some modems can be ordered with a built-in TDM.

**Statistical TDM (Statmux).** A statmux is more complex than a TDM. Allocating its capacity dynamically among users, a statmux collects incoming data into blocks and then transmits the blocks using an error-detecting-and-correcting protocol. Forming blocks slows transmission slightly, but using a statmux provides certain advantages:

- There are no errors on the link between statmuxes.
- Capacity is dynamically allocated to users actually sending data.
- Total apparent capacity may exceed the physical capacity for short periods because of buffering.
- Overhead is decreased because the statmuxes communicate using 8-bit bytes and data compression instead of the standard asynchronous 10-bit bytes.

The statmux also has some disadvantages. Flow control must be provided, because the statmux occasionally fills its buffers and must then stop incoming data to avoid data loss. Synchronous protocols normally require a special interface board in the statmux.

Using a statmux with a character-based terminal, a Tandem intelligent controller on the host, and XON/XOFF flow control is not a problem. However, using a statmux with a Tandem 6530 terminal can be difficult, especially when advanced statmux features are used. Tandem 6530-type terminals incorporate automatic flow control and a proprietary error-correcting protocol. In addition, they can alter their protocol at any time by switching between conversational mode and block mode.

It is crucial that the transmission path between the terminal and the Tandem host be as transparent as possible. Most intelligent modems, statmuxes, and other buffered devices are not transparent, often because they include features that assist character-based terminals by providing functions that the Tandem 6530 terminals already perform. These devices may interfere with the Tandem protocol. In addition, the functions they provide are not usually as comprehensive as the built-in Tandem functions. For example, their error-correction protocols cannot cover the data link between the device and the terminal.

Therefore, configuration recommendations for Tandem 6530 terminals on a statmux or other intelligent communications device are complex. Using a Tandem 6600 Intelligent Cluster Controller or an X.3 PAD instead of, or in addition to, the statmux is preferable. These devices, discussed later in this article, automatically modify their operation when the 6530 terminal switches between conversational and block mode. Both the X.3 PAD and the 6600 cluster controller use standard synchronous protocols that can be handled easily by any synchronous statmux or communications device.

Multiplexers (both TDMs and statmuxes) can handle asynchronous IBM 3270 emulators, but a better solution for clustered terminals is the use of a cluster controller. The link between the host and the cluster controller can be carried by IBM-compatible synchronous protocol cards in a multiplexer.

### **Multidrop Connections and Cluster Controllers**

Multidrop connections allow multiple devices to share a single communications link, but each device must have a unique address. (See Figure 6.) Moreover, the device must recognize its address when selected by the host (to receive data) or polled (to be given the opportunity to transmit). Normally, only page-based terminals have the multidrop option. Both Tandem 6530 and IBM 3270 terminals can share a multi-dropped line using IBM's SNA protocol. Both Tandem 6530 and IBM 3270 terminals also have older, non-standard protocols (Tandem AM6520 access method and IBM binary synchronous) that do not allow IBM and Tandem systems to share the same line.



The AM6520 access method is a proprietary Tandem protocol for multidropped asynchronous or synchronous lines. The hardware to handle this access method was originally built into all Tandem 6530 terminals, but current Tandem 6530 terminals and standard 6530 terminal emulators do not handle it. The AM6520 access method is still supported by Tandem host products and is available in a 6530 terminal emulation package from a Tandem Alliance member. Printers are handled either by a printer option within the terminal or by the Tandem 6820 terminal cluster concentrator (TCC), which can handle the protocol for one or two serial printers.

The AM6520 access method for multidrop environments was replaced by the Tandem 6600 Intelligent Cluster Controller. This controller uses IBM's 3270 SNA protocol and can share a physical link with IBM 3270 cluster controllers. The 6600 cluster controller attaches to a Tandem host by using the SNAX/XF I/O process. Any type of Tandem 6530 terminal or emulator can connect to the 6600 cluster controller, because it delivers the data to the Tandem 6530 terminal using Tandem's standard ATP6100 terminal protocol for direct connections. Serial printers can be attached, and remote users can dial into the 6600 cluster controller for connection to a Tandem host.

IBM 3270 terminals use either the older IBM binary synchronous protocol or IBM's newer SNA protocol. The terminals connect to an IBM 3270 cluster controller by a high-speed coaxial cable, and the cluster controller connects to the multidrop line. Tandem's AM3270 access method I/O process handles the IBM binary synchronous protocol for 3270 terminals. Tandem's SNAX/XF I/O process handles 3270 terminals connected by the SNA protocol. Tandem's Pathway system, electronic mail facility, and page-mode text editor all handle the IBM 3270 data stream directly. The AM3270 access method and the SNAX/XF I/O process incorporate automatic data stream conversion software for all Tandem and user-supplied conversational-mode applications.

Emulated 3270 terminals connect easily to Tandem hosts by using SNA/SDLC links. An example is Tandem's SNA Gateway for DOS or for OS/2, often used for dial-in situations. The host computer sees a single (emulated) terminal connected to a single-port (emulated) 3270 controller. Other SNA Gateway configurations allow multiple workstations to run 3270 emulations that share a single gateway across a LAN. Tandem's SNAX/XF I/O process handles the host end of the unswitched or switched connection.

## **X.25 Direct Connections**

An X.3 PAD<sup>2</sup> or IBM 3270 cluster controller with PAD capabilities is an excellent remote concentrator for access to a Tandem host. (See Figure 6.) A PAD is a microcomputer programmed to mediate between a character-based terminal and the X.25 packet interface. A typical PAD can be configured to handle up to 32 terminals concurrently. Some PADs are available on PC cards that can work with terminal emulators on the PC. Most X.3 PADs and 3270 cluster controller emulators can be configured to accept dial-in requests from terminals, automatically creating a connection to the Tandem host. If 6530 block mode must be emulated on ANSI 3.64 or DEC VT-type terminals, a combined PAD and TCP/IP terminal server running the TN3270 protocol can be used.

---

<sup>2</sup> X.3 is the international PAD standard set by the CCITT.

Tandem's X.25 Access Method (X25AM) I/O process is easily configured to allow a direct connection from the Tandem host to an X.3 PAD or PAD emulator, handling character-based, Tandem 6530, and IBM 3270 terminals. To communicate with IBM cluster controllers, a Tandem host requires both the SNAX/XF and X25AM I/O processes. TN3270 requires the TN32SERV, TCP/IP, and X25AM I/O processes. As discussed in the multidrop section, Tandem applications generally handle the IBM 3270 data stream directly.

### **Comparison of Group Connection Alternatives**

The three group connection types are markedly different from each other. Therefore, they must be evaluated separately.

Multiplexers are usually the least desirable design alternative. Unless suitable statmuxes are used and configured, performance with Tandem 6530 terminals may be poor, and random connection failures frequent. If different protocols must be combined over one physical

link (the one outstanding advantage of multiplexers), a reasonable design is to use another clustering method (cluster controller or X.25) and multiplex the communications line between the clustering device and the host instead of multiplexing the line directly connected to the terminal. This configuration eliminates the problem of 6530 terminal failures.

Multidrop or cluster controllers decrease the number of host ports needed to support the terminals. This option is cost-effective when more than one terminal is at each physical location. A cluster controller, which normally supports only page-based terminals, is usually the best clustering device. Tandem supports both the Tandem 6600 (for Tandem 6530 terminals) and the IBM 3270 cluster controller. As with all polled protocols, cluster controllers are an additional expense, but they can be justified easily for multiple terminals because of the savings in physical line costs. A rarely-used variation of this design, multidrop with a separate telephone line to each terminal, is very expensive (because of current communications tariffs) and is not discussed in this article.

X.25 designs are also a good choice in most cases, and they support all terminal types; however, X.25 has two major drawbacks. Printer management is less convenient than on the 6600 cluster controller. Also, unless the user buys an X.25 switch to handle PAD addressing, only one X.25 PAD can be configured on each access line to the Tandem host. Because X.25 switches are relatively inexpensive, they are preferable to multiplexers unless different protocols must be combined over one physical link. Direct X.3 PAD connections are best when only one remote group exists per physical link. Different types of terminals can be mixed on that link.

## Direct Remote Connections

Single terminals or workstations use modem connections for remote access. The physical connection between the terminal and the host can be permanent (a nonswitched or leased line), or it can be temporary for each use (a switched or dial-in line). In both cases, the modem uses Tandem's EIA-232 connection. Tandem's host I/O characteristics for modem connections are the same as those for local connections, with the exception of methods for changing line speed and for dialing switched lines. Figure 7 is an example of a direct remote connection.

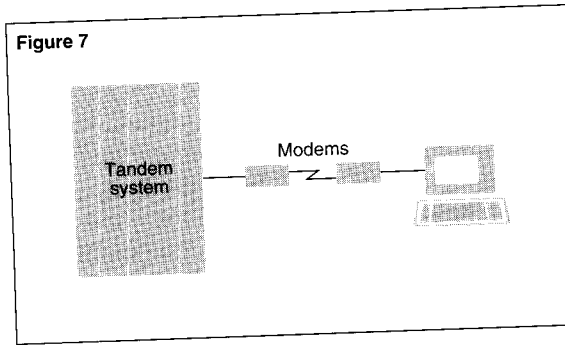
### Evaluation

The evaluation of this connection design is very similar to that of direct local connections. However, if Tandem 6530 terminals are used, speed-changing modems must be carefully chosen. Also, enhanced modems, which can decrease effective error rates for character-based terminals not made by Tandem, may interfere with Tandem 6530 terminals operating in page mode. If automatic dialing is required, the best interface is normally a second EIA-232 connection. EIA-366 is normally a poor choice in all situations, except when a large investment in old-style autodialers and software has already been made. The AT interface (originated and copyrighted by Hayes) mixes terminal data and modem commands on the same physical interface and requires custom software.

### Multispeed and Intelligent Modems

Asynchronous modems do not have a synchronizing wire in the EIA-232 interface; therefore, the user must provide a method for synchronizing line-speed changes between terminal and host. Two commonly used methods are a speed signal wire in the EIA-232 interface and a system for buffering data in the modem. These methods allow data to be received from and transmitted to the host at a fixed speed.

Figure 7



Tandem controllers can be configured with the DUAL212 option to respond to the speed wire used by AT&T 212 (and compatible) modems to switch between 1200 and 300 bits per second. However, AT&T 212 modems are rapidly becoming obsolete, replaced by triple-speed modems operating at 2400, 1200, and 300 bits per second and by V.32 and V.32bis modems that can operate at speeds of up to 14.4 kilobits per second. These higher-speed modems use buffering to maintain a constant speed on the EIA-232 host interface.

Figure 7.

Direct remote connection (switched or unswitched).

Character-based terminals using XON/XOFF flow control work well with the buffered modems and with Tandem, as they do not have a host-to-terminal protocol. The enhanced facilities typically included in buffered modems (such as error and flow control) are useful for character-based terminals. However, they cannot match the built-in Tandem 6530 facilities, because the link between the terminal or host and the modem is unprotected.

For Tandem 6530 terminals, transmission must be completely transparent, and the working configuration must be set into the modem permanently. For example, a standard V.32 modem works very well with a Tandem 6530 terminal. However, a V.32 modem enhanced with flow control (not needed by a 6530) may cause communications failure by interfering with the 6530's protocol.

If modem flow control is required, use intelligent controllers and hardware flow control at the Tandem host. Current 6530 terminals can also handle hardware flow control. Do not use XON/XOFF flow control at either the host or the terminal. For error correction and data compression, use V.42 and V.42bis; do not use MNP (Microcom Networking Protocol) Class 5.

For both character-based and Tandem 6530 terminals, Guardian 90 can control automatic answering if the modem is configured properly. The modem should answer only when the EIA-232 Data Terminal Ready (DTR) interface signal is present, and it should generate the EIA-232 Data Set Ready (DSR) interface signal when it is online.

### **Autodialers**

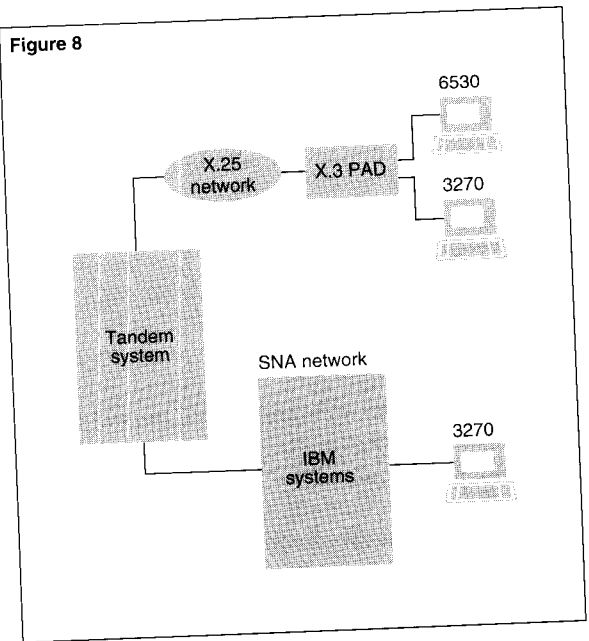
If they only receive calls, modems for switched lines are virtually identical to modems for unswitched lines. Complexity may arise only when the modem must dial to originate a switched call.

Early dialing modems used the AT&T 801 autodialer with the EIA-366 (V.25) interface. Tandem handles that interface with a 6303, 6304, or 3603 Envoy asynchronous I/O controller. Each EIA-366 interface requires two I/O ports and a special Y-shaped cable.

Because of the high cost of the I/O controller interface on many mainframe manufacturers' hosts, the industry has evolved toward either using a second EIA-232 port or embedding dialing information in the data stream itself. Using a second EIA-232 port is normally much less expensive than buying a controller for the special RS-366 interface, and the port is handled by Guardian 90 WRITEREAD commands. Most newly-manufactured autodialers can be configured to use either EIA-232 or EIA-366. Adapters (such as Racal-Vadic's VA831) are available to convert EIA-232 into EIA-366 for older autodialers.

Using the data interface to carry dialing information is less expensive than purchasing a second EIA-232 port; however, the complexity of that approach causes some problems. The method is only partially standardized, and the dialing information for the modem must be kept separate from the data stream sent to the terminal.

One of the most common dialing interfaces is the Hayes AT command set, which introduces problems caused by its mixed data-and-command interface. A standard modem, which does not mix data and commands, indicates to the host that it is ready to carry data by turning on the EIA-232 interface DSR signal wire. A modem using the AT commands may require that the host read the ASCII response from the modem in the data stream (for example, CONNECT). Also, some character patterns (for example, +++) affect the operation of the modem by placing it in command mode. Tandem does not supply software to handle these mixed data-and-command interfaces, but they can be built by the user.



**Figure 8.**  
WAN connection.

## Wide Area Network Communications

Wide area networks (WANs) connect geographically separate locations into a single system. WAN costs are computed differently than dedicated link costs, usually by number of characters transferred instead of by distance and time. Therefore, WANs may cost less than dedicated links for certain applications. The two most commonly used networks are X.25 and IBM's SNA. Tandem offers sophisticated facilities for connection to both of them. The TCP/IP protocol suite uses X.25 as a WAN network layer. Figure 8 is an example of a WAN connection.

### X.25 WAN Communications

The evaluation of networked X.25 connections is virtually identical to the evaluation of direct X.25 connections, described earlier in this article. The primary difference is the increased delay on the X.25 link, which may cause significant problems in page-mode operation. A generally preferable alternative is to use the Tandem SNA Gateway. The SNA Gateway uses SNA across the X.25 network instead of the Tandem 6530 protocol, avoiding the 6530 protocol's wait-for-acknowledgment requirement.

The X.25 protocol can be used for 6530 terminal access across a real X.25 network in addition to access over a direct link, as described previously. The transmission delay, which is negligible in the directly attached case, becomes an important factor in a WAN. Page transmission is markedly slower than with direct links, because the Tandem 6530 terminal always waits for an acknowledgement after transmitting a block. Data traffic should be minimized by using Pathway's ability to store pages in the 6530 terminal. To speed up transmission of filled pages, the block size (configurable on all current Tandem terminals and in the X25AM host I/O process) should be set to a large multiple of the X.25 packet size (for example, 512 bytes). Whenever possible, distributed systems should place the X.25 link between requester and server application processes instead of between the terminal and the requester process.

Although the TCP/IP suite of protocols can be used over an X.25 connection, such a situation is unusual, adding considerable overhead to the connection without improving Tandem 6530 terminal operation. This configuration does help decrease the error rate of character-based terminals, and it may be necessary if the user and the Tandem host are in separate networks bridged only by the Internet protocol. The TCP/IP Telnet NVT can be handled by Tandem as a simple character-based terminal, but it cannot display block-mode Pathway screens. A formatted character-based terminal (for example, a DEC VT-100) can be carried by the TCP/IP Telnet protocol, but the Tandem host needs special software to allow simulation of a block-mode 6530 screen unless the terminal's terminal server uses the TN3270 protocol over TCP/IP. In that case, the Guardian 90 TN3270 server can be used to display block-mode Pathway screens.

## **SNA WAN Communications**

Terminals that cross an SNA network from one SNA host to another before reaching the Tandem host require special attention. Complicated actions must be performed in the SNA environment to attach a terminal on one host to an application in a different host.

Tandem hosts running the SNAX/CDF (Systems Network Architecture Communications Services/Cross-Domain Facility) I/O software can connect to processes and terminals associated with an IBM host and using LU types 0, 1, 2, 3, 4, 6.2, and 7. Therefore, any terminal or workstation attached to any part of an SNA network can use that network to reach the Tandem host as though the terminal were directly attached. A special portion of the SNAX/CDF software (the Creator), allows the Tandem system to act as a true IBM applications host, complete with methods for starting processes or Pathway threads to accommodate new connections from elsewhere in the SNA network. Tandem applications generally handle the IBM 3270 data stream directly.

SNAX/CDF software is the only solution supplied by Tandem for attaching a Tandem system to an existing SNA network to receive terminal connections from that network using standard SNA architecture. However, Tandem Alliance partners provide other solutions that provide different sets of features, some based on Node Type 2.1 to provide full terminal connectivity in a nonstandard way.

---

## Conclusion

The connection design most appropriate to user requirements depends on the user's terminal types, applications, and existing facilities. In general, dial-in and locally attached asynchronous lines are the simplest, but clustered communications using multidrop facilities or X.25 direct connections can provide communications cost savings with a slight increase in complexity and host overhead. LAN-based connections are versatile; they may constitute the best design for a specific situation despite their configuration and management complexities. WANs (such as X.25 and SNA) are usually specified for widely distributed user bases having low-to-moderate volume.

For local distributions, the preferred design is the inexpensive, easy-to-manage direct local connection. For campus distributions, the preferred design is either LAN or group. Grouped remote distributions are usually handled by grouped designs, but bridges to remote LANs are becoming more common. Isolated remote distributions are handled best by either direct remote or WAN connections, depending primarily on cost and performance considerations.

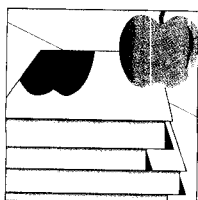
## Acknowledgments

The author would like to thank Dave Mathes, Mike McDonald, Justin Simonds, Janet Smith, and Bob Stettina for their work reviewing the article.

---

**Eric D. Siegel** joined Tandem in 1984 and is currently with the Integration and Technology Center, where he provides consulting services to Tandem customers. He wrote the core data communications course for Tandem analysts and is the corporate focal point for all data communications specialists. Eric has BS and MEng degrees in Electrical Engineering from Cornell University.

# Technical Information & Education



## Professional Services

Tandem Professional Services is a program in which trained Tandem experts deliver technical consulting services at the user site. Tandem now offers a number of standardized services and will announce additional ones in this department of the *Tandem Systems Review* as they become available. For more information, users should contact their local Tandem representative.

### Security Review Service

*May 1992*

This service assesses the security of the user's existing Tandem NonStop system. Its objective is to provide a clear description of the strengths and potential exposures of the user's Tandem security implementation. As part of this service, Tandem security specialists work with the user's personnel to gather and analyze the relevant data and to recommend ways to enhance the level of security of the user's system.

## Technical Information Series Books

Tandem Technical Information Series (TIS) books are issued on a yearly subscription basis. Each book covers in depth a specific technical topic related to Tandem systems. The Series helps users reach the right technical decisions and solutions to meet business requirements. For more information on the TIS books, users should call 1-800-473-5868.

### Functional Comparison of Tandem's Pathway and TMF with IBM's IMS/ESA

*April 1992*

This book, the first in the TIS series, compares functionally similar online transaction processing (OLTP) products from IBM and Tandem. The IBM product is known as Information Management System/Enterprise Systems Architecture (IMS/ESA). Tandem's Pathway distributed transaction processing system and TMF (Transaction Monitoring Facility) software, taken together, are similar to the IBM product.

*The Technical Information and Education department is an annotated list of new consulting and information services, software education courses, and books Tandem is offering its users.*



The book aims to reduce the language barriers between IBM and Tandem specialists. To accomplish this objective, the book shows the similarities of and differences between the capabilities and terminologies of the IBM and Tandem systems.

The sections in this book present the architectural structures of the IBM and Tandem hardware and operating systems (including the available communications protocols) and relate the equivalent functions of both systems. The book's side-by-side format allows the reader to thumb through each section and immediately see the functional equivalents of the Pathway/TMF and IMS/ESA systems.

## CD Read Documentation

Tandem CD Read provides a complete set of Guardian 90 operating system software manuals on a single CD-ROM disc. Periodically, all CD Read subscribers receive a new disc containing the latest set of Guardian software manuals.

### **CD Read, Version C30\_07\_01** *March 1992*

Version C30\_07\_01 of CD Read is now available. Beginning with this version, CD Read includes softdocs.

## Software Education

The following paragraphs provide highlights of the latest software education courses offered by Tandem. To sign up for a class or to order an independent study program (ISP), users should call 1-800-621-9188. Full descriptions of all available courses and ISPs appear in the *Software Education Course Catalog* and on InfoWay.

### **Basic Tandem Operator Tasks**

This independent study program introduces new Tandem operators to NonStop system hardware and software. It also shows how to carry out basic operator tasks such as monitoring the system, running batch jobs, managing print jobs, and working with tapes.

### **COBOL Programming I**

This five-day course provides instruction in designing and writing COBOL programs using ANSI-standard COBOL85. The course presents an introduction to the COBOL language including basic concepts, syntax, and program design considerations. Along with designing and writing COBOL programs, students use case statements, file sorting, file organization, and access modes. They also learn to use structured programming techniques.

### **COBOL Programming II: Tandem Extensions**

This five-day course introduces Tandem extensions and advanced techniques to the COBOL85 language. This class includes a review of all file types and access modes, as well as interprocess and interprogram communications. Students learn how to write requesters and servers in COBOL, use NonStop programming and TMF, and understand the major new features in COBOL85. A summary of COBOL 74 and COBOL85 differences is included as part of this course. Students also learn how to use subprograms and nested programs with global and local variables.

### **Enform**

This five-day course provides instruction and practice in the use of the Enform query language and report generator. The course format alternates between lectures that introduce Enform features and extensive labs that provide immediate hands-on experience.

### **NonStop SQL Basics**

This four-day course is a general introduction to NonStop SQL and serves as a prerequisite to more advanced courses. Hands-on lab sessions provide practical experience with NonStop SQL.

### **Pathway Application Programming Education Series**

The Pathway Application Programming Education Series offers a quick and efficient way to learn to develop Pathway applications. Through hands-on activities, students gain and demonstrate practical application programming skills in the Pathway environment.

The series includes the following independent study programs:

- SCREEN COBOL Requesters.
- Server Fundamentals.
- Servers That Access Enscribe Databases.
- Servers That Access NonStop SQL Databases.
- Pathway Configuration and Operations.

## **Security for Auditors**

Security for Auditors presents to EDP auditors the Tandem Guardian 90 system concepts, with emphasis on system, subsystem, and application security features. This five-day course provides an introduction to the tools for auditing a Tandem system.

## **Tandem Open Systems Interconnection Education Series**

The Tandem Open Systems Interconnection (OSI) Education Series teaches students how to plan, install, and configure a Tandem OSI end system with Tandem OSI Transport Services (OSI/TS) or Tandem OSI Application Services (OSI/AS). This series includes six independent study programs (ISPs). Users can complete all six ISPs in sequence or complete one ISP at a time in the order and combination that suits their particular job needs. If the user installation is large enough, it may be appropriate to divide the task among team members and assign appropriate ISP subsets to each team member.

The series includes these ISPs:

- Tandem OSI Product Introduction.
- OSI Naming and Addressing Concepts.
- X25AM SYSGEN and TEST.
- TLAM SYSGEN and TEST.
- OSI Transport Services (OSI/TS) Installation and Configuration.
- OSI Application Services (OSI/AS) Installation and Configuration.

## InfoWay Online Support Service

**C**ustomer support has always been of paramount importance at Tandem™. A key element of this support is the flow of information and resources between Tandem, its customers, and its partners. To facilitate and enhance this aspect of customer support, Tandem has developed an online service called InfoWay™.

InfoWay gives subscribers direct electronic access to their account team, various product and support databases, and a growing number of support services. Tandem offers current InfoWay services to all customers and Tandem Alliance partners at no charge, in order to help Tandem users manage their systems more effectively.

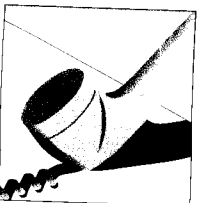
### Current Services

InfoWay users at present have access to a variety of information and communication services. Tandem will continue to make additional services available to InfoWay users.

### Electronic Mail

Through InfoWay, subscribers can communicate with their account team by way of electronic mail. (Tandem Alliance partners can exchange electronic mail with anyone at Tandem.) In addition, all subscribers can join special interest groups (SIGs), groups of subscribers sharing a common interest in a business topic oriented toward Tandem. Members of a SIG can broadcast electronic mail messages to the group, share experiences, and ask for or offer advice on specific issues.

Electronic mail has proven effective at speeding the diagnosis of system or program problems. Users can quickly send files containing symptoms to their support team for first-hand diagnosis and resolution of problems.



At present, subscribers in several countries are also able to use electronic mail to communicate with their Tandem NonStop Support Center (TNSC) representatives. Tandem is expanding this service on a country-by-country basis. Problem reports to the TNSC must still be initiated by telephone, but the follow-up information can be exchanged electronically as this capability becomes available in more countries.

### **System Software Inventory**

InfoWay makes it possible for users to conduct a comprehensive inventory of their Tandem system software. Users can determine which software versions they have, and which enhancements, or interim product modifications (IPMs), they lack and may want to install.

This service, called software version analyzer (SVA), keeps users aware of the specific IPMs that apply to their configuration. SVA also helps users verify that they are meeting an IPM's prerequisites.

### **IPM Information**

InfoWay users can review and evaluate in depth any IPM. They can access and download IPM information, software alerts, bulletins, and techniques for getting around operating problems. This helps users prevent problems, use products more effectively, and determine which IPMs or techniques are appropriate for them. Also, users may ask to be notified, once a month, of all the IPMs released that month.

### **Access to Documentation**

Through InfoWay, users can read and download software installation documentation (softdocs), release installation documentation, product maintenance information, and other relevant technical information. All of this information helps in planning an error-free software installation.

In addition, InfoWay users are able to

- Review the training class schedules and descriptions for most of the Tandem Education Centers around the world.

- Read extracts from a number of publications such as the *Tandem Systems Review* and the *ITUG Journal*.

- Read and download product marketing literature such as datasheets, brochures, and press releases.

### **Tandem Alliance Directory**

InfoWay gives users access to the Tandem Alliance Directory. This resource provides InfoWay users with extensive information about the services and solutions available from Tandem Alliance partners and consultants. Users can review summary descriptions of the partners' products, company histories, and contacts.

### **InfoWay Online Features and Services**

InfoWay subscribers can

- Use electronic mail to communicate with their account team.
- Review and evaluate individual product enhancements in depth.
- Access extensive software product and support documentation.
- Review Tandem training class schedules.
- Read extracts from publications such as the *Tandem Systems Review* and the *ITUG Journal*.
- Read product marketing literature.
- Access information on Tandem Alliance solutions.

---

## Using InfoWay

InfoWay is available to subscribers 24 hours a day, 7 days a week. InfoWay is menu-driven and provides extensive online help. All subscribers also receive a printed copy of the *InfoWay User's Guide* when they sign up for the service.

For their InfoWay sessions, users can choose either of two user interfaces, a conversational interface or a block-mode interface. After establishing their session, users select the service or resource they want, complete their task, and dissolve the session.

A highly useful feature of InfoWay is its phrase search capability. Users can search InfoWay databases by using a phrase such as "bank transaction messages" or "Cyclone system tuning."

InfoWay resides at three Tandem service points, located in Cupertino, California; Frankfurt, Germany; and Singapore. Users may access the service point most convenient to them and incur communications costs only to that service point.

A user can access InfoWay from a Tandem system terminal or from a stand-alone PC or workstation over a public data network. However, access from a Tandem system provides a wider and better choice of services. Furthermore, many future services will require access from a Tandem system. Access can be asynchronous or by way of X.25.

InfoWay access from the user's Tandem system is fully secure. The user's system manager can control system access to InfoWay and InfoWay access to the user's system. Moreover, InfoWay runs as a nonprivileged process, which further guarantees security.

Users can sign up for access to InfoWay either through their account team or TNSC, or by filling out a form provided on their site update tapes. To learn more about InfoWay, users should contact their local Tandem representative.

# TandemSystemsReviewIndex

The *Tandem Journal* became the *Tandem Systems Review* in February 1985. Four issues of the *Tandem Journal* were published:

Volume 1, Number 1	Fall 1983	Part no. 83930
Volume 2, Number 1	Winter 1984	Part no. 83931
Volume 2, Number 2	Spring 1984	Part no. 83932
Volume 2, Number 3	Summer 1984	Part no. 83933

As of this issue, 18 issues of the Tandem Systems Review have been published:

Volume 1, Number 1	February 1985	Part no. 83934
Volume 1, Number 2	June 1985	Part no. 83935
Volume 2, Number 1	February 1986	Part no. 83936
Volume 2, Number 2	June 1986	Part no. 83937
Volume 2, Number 3	December 1986	Part no. 83938
Volume 3, Number 1	March 1987	Part no. 83939
Volume 3, Number 2	August 1987	Part no. 83940
Volume 4, Number 1	February 1988	Part no. 11078
Volume 4, Number 2	July 1988	Part no. 13693
Volume 4, Number 3	October 1988	Part no. 15748
Volume 5, Number 1	April 1989	Part no. 18662
Volume 5, Number 2	September 1989	Part no. 28152
Volume 6, Number 1	March 1990	Part no. 32986
Volume 6, Number 2	October 1990	Part no. 46987
Volume 7, Number 1	April 1991	Part no. 46988
Volume 7, Number 2	October 1991	Part no. 65248
Volume 8, Number 1	Spring 1992	Part no. 65250
Volume 8, Number 2	Summer 1992	Part no. 69848

The articles published in all 22 issues are arranged by subject below. (*Tandem Journal* is abbreviated as TJ and *Tandem Systems Review* as TSR.) A second index, arranged by product, is also provided.

## Index by Subject

Article title	Author(s)	Publication	Volume, Issue	Publication date	Part number
<b>APPLICATION DEVELOPMENT AND LANGUAGES</b>					
Ada: Tandem's Newest Compiler and Programming Environment	R. Vnuk	TSR	3,2	Aug. 1987	83940
A New Design for the PATHWAY TCP	R. Wong	TJ	2,2	Spring 1984	83932
An Introduction to Tandem EXTENDED BASIC	J. Meyerson	TJ	2,2	Spring 1984	83932
Debugging TAOL Code	L. Palmer	TSR	4,2	July 1988	13693
Instrumenting Applications for Effective Event Management	J. Dagenais	TSR	7,2	Oct. 1991	65248
New TAL Features	C. Lu, J. Murayama	TSR	2,2	June 1986	83837
PATHFINDER—An Aid for Application Development	S. Benett	TJ	1,1	Fall 1983	83930
PATHWAY IDS: A Message-level Interface to Devices and Processes	M. Anderton, M. Noonan	TSR	2,2	June 1986	83937

Article title	Author(s)	Publication	Volume, Issue	Publication date	Part number
<b>APPLICATION DEVELOPMENT AND LANGUAGES (cont.)</b>					
State-of-the-Art C Compiler	E. Kit	TSR	2,2	June 1986	83937
TACL, Tandem's New Extensible Command Language	J. Campbell, R. Glascock	TSR	2,1	Feb. 1986	83936
Tandem's New COBOL85	D. Nelson	TSR	2,1	Feb. 1986	83936
The ENABLE Program Generator for Multifile Applications	B. Chapman, J. Zimmerman	TSR	1,1	Feb. 1985	83934
TMF and the Multi-Threaded Requester	T. Lemberger	TJ	1,1	Fall 1983	83930
Writing a Command Interpreter	D. Wong	TSR	1,2	June 1985	83935
<b>CUSTOMER SUPPORT</b>					
Customer Information Service	J. Massucco	TSR	3,1	March 1987	83939
Remote Support Strategy	J. Eddy	TSR	3,1	March 1987	83939
Tandem's Software Support Plan	R. Baker, D. McEvoy	TSR	3,1	March 1987	83939
<b>DATA COMMUNICATIONS</b>					
An Overview of SNAX/CDF	M. Turner	TSR	5,2	Sept. 1989	28152
A SNAX Passthrough Tutorial	D. Kirk	TJ	2,2	Spring 1984	83932
Changes in FOX	N. Donde	TSR	1,2	June 1985	83935
Connecting Terminals and Workstations to Guardian 90 Systems	E. Siegel	TSR	8,2	Summer 1992	69848
Introduction to MULTILAN	A. Coyle	TSR	4,1	Feb. 1988	11078
Overview of the MULTILAN Server	A. Rowe	TSR	4,1	Feb. 1988	11078
SNAX/APC: Tandem's New SNA Software or Distributed Processing	B. Grantham	TSR	3,1	March 1987	83939
SNAX/HLS: An Overview	S. Saltwick	TSR	1,2	June 1985	83935
TLAM: A Connectivity Option for Expand	K. MacKenzie	TSR	7,1	April 1991	46988
Using the MULTILAN Application Interfaces	M. Berg, A. Rowe	TSR	4,1	Feb. 1988	11078
<b>DATA MANAGEMENT</b>					
A Comparison of the B00 DP1 and DP2 Disc Processes	T. Schachter	TSR	1,2	June 1985	83935
An Overview of NonStop SQL Release 2	M. Pong	TSR	6,2	Oct. 1990	46987
Batch Processing in Online Enterprise Computing	T. Keefauver	TSR	6,2	Oct. 1990	46987
Concurrency Control Aspects of Transaction Design	W. Senf	TSR	6,1	March 1990	32968
Converting Database Files from ENSCRIBE to NonStop SQL	W. Weikel	TSR	6,1	March 1990	32986
DP1-DP2 File Conversion: An Overview	J. Tate	TSR	2,1	Feb. 1986	83936
Determining FCP Conversion Time	J. Tate	TSR	2,1	Feb. 1986	83936
DP2's Efficient Use of Cache	T. Schachter	TSR	1,2	June 1985	83935
DP2 Highlights	K. Carlyle, L. McGowan	TSR	1,2	June 1985	83935
DP2 Key-sequenced Files	T. Schachter	TSR	1,2	June 1985	83935
Gateways to NonStop SQL	D. Slutz	TSR	6,2	Oct. 1990	46987
High-Performance SQL Through Low-Level System Integration	A. Borr	TSR	4,2	July 1988	13693
Improvements in TMF	T. Lemberger	TSR	1,2	June 1985	83935
Online Reorganization of Key-Sequenced Tables and Files	G. Smith	TSR	6,2	Oct. 1990	46987



Article title	Author(s)	Publication	Volume, Issue	Publication date	Part number
<b>DATA MANAGEMENT (cont.)</b>					
Optimizing Batch Performance	T. Keefauver	TSR	5,2	Sept. 1989	28152
Overview of NonStop SQL	H. Cohen	TSR	4,2	July 1988	13693
Parallelism in NonStop SQL Release 2	M. Moore, A. Sodhi	TSR	6,2	Oct. 1990	46987
NetBatch: Managing Batch Processing on Tandem Systems	D. Wakashige	TSR	5,1	April 1989	18662
NetBatch-Plus: Structuring the Batch Environment	G. Earle, D. Wakashige	TSR	6,1	March 1990	32986
NonStop SQL: The Single Database Solution	J. Cassidy, T. Kocher	TSR	5,2	Sept. 1989	28152
NonStop SQL Data Dictionary	R. Holbrook, D. Tsou	TSR	4,2	July 1988	1369368
NonStop SQL Optimizer: Basic Concepts	M. Pong	TSR	4,2	July 1988	13693
NonStop SQL Optimizer: Query Optimization and User Influence	M. Pong	TSR	4,2	July 1988	13693
NonStop SQL Reliability	C. Fenner	TSR	4,2	July 1988	13693
The NonStop SQL Release 2 Benchmark	S. Englert, J. Gray, T. Kocher, P. Shah	TSR	6,2	Oct. 1990	46987
The Outer Join in NonStop SQL	J. Vaishnav	TSR	6,2	Oct. 1990	46987
The Relational Data Base Management Solution	G. Ow	TJ	2,1	Winter 1984	83931
Tandem's NonStop SQL Benchmark	Tandem Performance Group	TSR	4,1	Feb. 1988	11078
The TRANSFER Delivery System for Distributed Applications	S. Van Pelt	TJ	2,2	Spring 1984	83932
TMF Autorollback: A New Recovery Feature	M. Pong	TSR	1,1	Feb. 1985	83934
<b>MANUALS/COURSES</b>					
B00 Software Manuals	S. Olds	TSR	1,2	June 1985	83935
C00 Software Manuals	E. Levi	TSR	4,1	Feb. 1988	11078
New Software Courses	M. Janow	TSR	1,2	June 1985	83935
New Software Courses	J. Limper	TSR	4,1	Feb. 1988	11078
Subscription Policy for Software Manuals	T. McSweeney	TSR	2,1	Feb. 1986	83936
Tandem's New Products	C. Robinson	TSR	2,1	Feb. 1986	83936
Tandem's New Products	C. Robinson	TSR	2,2	June 1986	83937
<b>OPERATING SYSTEMS</b>					
Highlights of the B00 Software Release	K. Coughlin, R. Montevaldo	TSR	1,2	June 1985	83935
Increased Code Space	A. Jordan	TSR	1,2	June 1985	83935
Managing System Time Under GUARDIAN 90	E. Nellen	TSR	2,1	Feb. 1986	83936
New GUARDIAN 90 Time-keeping Facilities	E. Nellen	TSR	1,2	June 1985	83935
New Process-timing Features	S. Sharma	TSR	1,2	June 1985	83935
NonStop II Memory Organization and Extended Addressing	D. Thomas	TJ	1,1	Fall 1983	83930
Overview of the C00 Release	L. Marks	TSR	4,1	Feb. 1988	11078
Overview of the NonStop-UX Operating System for the Integrity S2	P. Norwood	TSR	7,1	April 1991	46988
Robustness to Crash in a Distributed Data Base: A Nonshared-memory Approach	A. Borr	TSR	1,2	June 1985	83935
The GUARDIAN Message System and How to Design for It	M. Chandra	TSR	1,1	Feb. 1985	83935
The Tandem Global Update Protocol	R. Carr	TSR	1,2	June 1985	83935

Article title	Author(s)	Publication	Volume, Issue	Publication date	Part number
<b>PERFORMANCE AND CAPACITY PLANNING</b>					
A Performance Retrospective	P. Oleinick, P. Shah	TSR	2,3	Dec. 1986	83938
Buffering for Better Application Performance	R. Mattran	TSR	2,1	Feb. 1986	83936
Capacity Planning Concepts	R. Evans	TSR	2,3	Dec. 1986	83938
Capacity Planning With TCM	W. Highleyman	TSR	7,2	Oct. 1991	65248
C00 TMDS Performance	J. Mead	TSR	4,1	Feb. 1988	11078
Credit-authorization Benchmark for High Performance and Linear Growth	T. Chmiel, T. Houy	TSR	2,1	Feb. 1986	83936
Debugging Accelerated Programs on TNS/R Systems	D. Cressler	TSR	8,1	Spring 1992	65250
DP2 Performance	J. Enright	TSR	1,2	June 1985	83935
Estimating Host Response Time in a Tandem System	H. Horwitz	TSR	4,3	Oct. 1988	15748
FASTSORT: An External Sort Using Parallel Processing	J. Gray, M. Stewart, A. Tsukerman, S. Uren, B. Vaughan	TSR	2,3	Dec. 1986	83938
Getting Optimum Performance from Tandem Tape Systems	A. Khatri	TSR	2,3	Dec. 1986	83938
How to Set Up a Performance Data Base with MEASURE and ENFORM	M. King	TSR	2,3	Dec. 1986	83938
Improved Performance for BACKUP2 and RESTORE2	A. Khatri, M. McCline	TSR	1,2	June 1985	83935
Improving Performance on TNS/R Systems With the Accelerator	M. Blanchet	TSR	8,1	Spring 1992	65250
MEASURE: Tandem's New Performance Measurement Tool	D. Dennison	TSR	2,3	Dec. 1986	83938
Measuring DSM Event Management Performance	M. Stockton	TSR	8,1	Spring 1992	65250
Message System Performance Enhancements	D. Kinkade	TSR	2,3	Dec. 1986	83938
Message System Performance Tests	S. Uren	TSR	2,3	Dec. 1986	83938
Network Design Considerations	J. Evjen	TSR	5,2	Sept. 1989	28152
NonStop VLX Performance	J. Enright	TSR	2,3	Dec. 1986	83938
Optimizing Sequential Processing on the Tandem System	R. Welsh	TJ	2,3	Summer 1984	83933
Pathway TCP Enhancements for Application Run-Time Support	R. Vannucci	TSR	7,1	April 1991	46988
Performance Benefits of Parallel Query Execution and Mixed Workload Support in NonStop SQL Release 2	S. Englert, J. Gray	TSR	6,2	Oct. 1990	46987
Performance Considerations for Application Processes	R. Glasstone	TSR	2,3	Dec. 1986	83938
Performance Measurements of an ATM Network Application	N. Cabell, D. Mackie	TSR	2,3	Dec. 1986	83938
Predicting Response Time in On-line Transaction Processing Systems	A. Khatri	TSR	2,2	June 1986	83937
The 6600 and TCC6820 Communications Controllers: A Performance Comparison	P. Beadles	TSR	2,3	Dec. 1986	83938
The ENCORE Stress Test Generator for On-line Transaction Processing Applications	S. Kosinski	TJ	2,1	Winter 1984	83931
The PATHWAY TCP: Performance and Tuning	J. Vatz	TSR	1,1	Feb. 1985	83934
The Performance Characteristics of Tandem NonStop Systems	J. Day	TJ	1,1	Fall 1983	83930
Sizing Cache for Applications that Use B-series DP1 and TMF	P. Shah	TSR	2,2	June 1986	83937
Sizing the Spooler Collector Data File	H. Norman	TSR	4,1	Feb. 1988	11978

Article title	Author(s)	Publication	Volume, Issue	Publication date	Part number
<b>PERFORMANCE AND CAPACITY PLANNING (cont.)</b>					
Tandem's 5200 Optical Storage Facility: Performance and Optimization Considerations	S. Coleman	TSR	5,1	April 1989	18662
Tandem's Approach to Fault Tolerance	B. Bail, W. Bartlett, S. Thompson	TSR	4,1	Feb. 1988	11078
Understanding PATHWAY Statistics	R. Wong	TJ	2,2	Spring 1984	83932
<b>PERIPHERALS</b>					
5120 Tape Subsystem Recording Technology	W. Phillips	TSR	3,2	Aug. 1987	83940
An Introduction to DYNAMITE Workstation Host Integration	S. Kosinski	TSR	1,2	June 1985	83935
Data-Encoding Technology Used in the XL8 Storage Facility	D.S. Ng	TSR	2,2	June 1986	83937
Data-Window Phase-Margin Analysis	A. Painter, H. Pham, H. Thomas	TSR	2,2	June 1986	83937
Introducing the 3207 Tape Controller	S. Chandran	TSR	1,2	June 1985	83935
Peripheral Device Interfaces	J. Blakkan	TSR	3,2	Aug. 1987	83940
Plated Media Technology Used in the XL8 Storage Facility	D.S. Ng	TSR	2,2	June 1986	83937
Streaming Tape Drives	J. Blakkan	TSR	3,2	Aug. 1987	83940
Terminal Selection	E. Siegel	TSR	8,2	Summer 1992	69848
The 5200 Optical Storage Facility: A Hardware Perspective	A. Patel	TSR	5,1	April 1989	18662
The 6100 Communications Subsystem: A New Architecture	R. Smith	TJ	2,1	Winter 1984	83931
The 6600 and TCC6820 Communications Controllers: A Performance Comparison	P. Beadles	TSR	2,3	Dec. 1986	83938
The DYNAMITE Workstation: An Overview	G. Smith	TSR	1,2	June 1985	83935
The Model 6VI Voice Input Option: Its Design and Implementation	B. Huggett	TJ	2,3	Summer 1984	83933
The Role of Optical Storage in Information Processing	L. Sabaroff	TSR	3,2	Aug. 1987	83940
The V8 Disc Storage Facility: Setting a New Standard for On-line Disc Storage	M. Whiteman	TSR	1,2	June 1985	83935
<b>PROCESSORS</b>					
Fault Tolerance in the NonStop Cyclone System	S. Chan, R. Jardine	TSR	7,1	April 1991	46988
NonStop CLX: Optimized for Distributed On-Line Transaction Processing	D. Lenoski	TSR	5,1	April 1989	18662
NonStop VLX Hardware Design	M. Brown	TSR	2,3	Dec. 1986	83938
Overview of Tandem NonStop Series/RISC Systems	L. Faby, R. Mateosian	TSR	8,1	Spring 1992	65250
The High-Performance NonStop TXP Processor	W. Bartlett, T. Houy, D. Meyer	TJ	2,1	Winter 1984	83931
The NonStop TXP Processor: A Powerful Design for On-line Transaction Processing	P. Oleinick	TJ	2,3	Summer 1984	83933
The VLX: A Design for Serviceability	J. Allen, R. Boyle	TSR	3,1	March 1987	83939
<b>SECURITY</b>					
Dial-In Security Considerations	P. Grainger	TSR	7,2	Oct. 1991	65248
Distributed Protection with SAFEGUARD	T. Chou	TSR	2,2	June 1986	83937
Enhancing System Security With Safeguard	C. Gaydos	TSR	7,1	April 1991	46988

Article title	Author(s)	Publication	Volume, Issue	Publication date	Part number
<b>SYSTEM CONNECTIVITY</b>					
Building Open Systems Interconnection with OSI/AS and OSI/TS	R. Smith	TSR	6,1	March 1990	32986
Connecting Terminals and Workstations to Guardian 90 Systems	E. Siegel	TSR	8,2	Summer 1992	69848
Network Design Considerations	J. Evjen	TSR	5,2	Sept. 1989	28152
Terminal Connection Alternatives for Tandem Systems	J. Simonds	TSR	5,1	April 1989	18662
Terminal Selection	E. Siegel	TSR	8,2	Summer 1992	69848
The OSI Model: Overview, Status, and Current Issues	A. Dunn	TSR	5,1	April 1989	18662
<b>SYSTEM MANAGEMENT</b>					
Configuring Tandem Disk Subsystems	S. Sittler	TSR	2,3	Dec. 1986	83938
Data Replication in Tandem's Distributed Name Service	T. Eastep	TSR	4,3	Oct. 1988	15748
Enhancements to TMDS	L. White	TSR	3,2	Aug. 1987	83940
Event Management Service Design and Implementation	H. Jordan, R. McKee, R. Schuet	TSR	4,3	Oct. 1988	15748
Introducing TMDS, Tandem's New On-line Diagnostic System	J. Troisi	TSR	1,2	June 1985	83935
Instrumenting Applications for Effective Event Management	J. Dagenais	TSR	7,2	Oct. 1991	65248
Measuring DSM Event Management Performance	M. Stockton	TSR	8,1	Spring 1992	65250
Network Statistics System	M. Miller	TSR	4,3	Oct. 1988	15748
Overview of DSM	P. Homan, B. Malizia, E. Reisner	TSR	4,3	Oct. 1988	15748
SCP and SCF: A General Purpose Implementation of the Subsystem Programmatic Interface	T. Lawson	TSR	4,3	Oct. 1988	15748
RDF: An Overview	J. Guerrero	TSR	7,2	Oct. 1991	65248
RDF Synchronization	F. Jongma, W. Senf	TSR	8,2	Summer 1992	69848
Tandem's Subsystem Programmatic Interface	G. Tom	TSR	4,3	Oct. 1988	15748
Using FOX to Move a Fault-tolerant Application	C. Breighner	TSR	1,1	Feb. 1985	83934
Using the Subsystem Programmatic Interface and Event Management Services	K. Stobie	TSR	4,3	Oct. 1988	15748
VIEWPOINT Operations Console Facility	R. Hansen, G. Stewart	TSR	4,3	Oct. 1988	15748
VIEWSYS: An On-line System-resource Monitor	D. Montgomery	TSR	1,2	June 1985	83935
Writing Rules for Automated Operations	J. Collins	TSR	7,2	Oct. 1991	65248)
<b>UTILITIES</b>					
Enhancements to PS MAIL	R. Funk	TSR	3,1	March 1987	839393

## Index by Products

Article title	Author(s)	Publication	Volume, Issue	Publication date	Part number
<b>3207 TAPE CONTROLLER</b>					
Introducing the 3207 Tape Controller	S. Chandran	TSR	1,2	June 1985	83935
<b>5120 TAPE SUBSYSTEM</b>					
5120 Tape Subsystem Recording Technology	W. Phillips	TSR	3,2	Aug. 1987	83940
<b>5200 OPTICAL STORAGE</b>					
Tandem's 5200 Optical Storage Facility: Performance and Optimization Considerations	S. Coleman	TSR	5,1	April 1989	18662
The 5200 Optical Storage Facility: A Hardware Perspective	A. Patel	TSR	5,1	April 1989	18662
The Role of Optical Storage in Information Processing	L. Sabaroff	TSR	4,1	Feb. 1988	11078
<b>6100 COMMUNICATIONS SUBSYSTEM</b>					
The 6100 Communications Subsystem: A New Architecture	R. Smith	TJ	2,1	Winter 1984	83931
<b>6530 TERMINAL</b>					
The Model 6VI Voice Input Option: Its Design and Implementation	B. Huggett	TJ	2,3	Summer 1984	83933
<b>6600 AND TCC6820 COMMUNICATIONS CONTROLLERS</b>					
The 6600 and TCC6820 Communications Controllers: A Performance Comparison	P. Beadles	TSR	2,3	Dec. 1986	83938
<b>Ada</b>					
Ada: Tandem's Newest Compiler and Programming Environment	R. Vnuk	TSR	3,2	Aug. 1987	83940
<b>BASIC</b>					
An Introduction to Tandem EXTENDED BASIC	J. Meyerson	TJ	2,2	Spring 1984	83932
<b>C</b>					
State-of-the-art C Compiler	E. Kit	TSR	2,2	June 1986	83937
<b>CIS</b>					
Customer Information Service	J. Massucco	TSR	3,1	March 1987	83939
<b>CLX</b>					
NonStop CLX: Optimized for Distributed On-Line Transaction Processing	D. Lenoski	TSR	5,1	April 1989	18662
<b>COBOL85</b>					
Tandem's New COBOL85	D. Nelson	TSR	2,1	Feb. 1986	83936
<b>COMINT (CI)</b>					
Writing a Command Interpreter	D. Wong	TSR	1,2	June 1985	83935
<b>CYCLONE</b>					
Fault Tolerance in the NonStop Cyclone System	S. Chan, R. Jardine	TSR	7,1	April 1991	46988
<b>DP1 AND DP2</b>					
A Comparison of the B00 DP1 and DP2 Disc Processes	T. Schachter	TSR	1,2	June 1985	83935
Determining FCP Conversion Time	J. Tate	TSR	2,1	Feb. 1986	83936
DP1-DP2 File Conversion: An Overview	J. Tate	TSR	2,1	Feb. 1986	83936
DP2 Highlights	K. Carlyle, L. McGowan	TSR	1,2	June 1985	83935
DP2 Key-sequenced Files	T. Schachter	TSR	1,2	June 1985	83935

Article title	Author(s)	Publication	Volume, Issue	Publication date	Part number
<b>DP1 AND DP2 (cont.)</b>					
DP2 Performance	J. Enright	TSR	1,2	June 1985	83935
DP2's Efficient Use of Cache	T. Schachter	TSR	1,2	June 1985	83935
Sizing Cache for Applications that Use B-series DP1 and TMF	P. Shah	TSR	2,2	June 1986	83937
<b>DSM</b>					
Data Replication in Tandem's Distributed Name Service	T. Eastep	TSR	4,3	Oct. 1988	15748
Event Management Service Design and Implementation	H. Jordan, R. McKee, R. Schuet	TSR	4,3	Oct. 1988	15748
Instrumenting Applications for Effective Event Management	J. Dagenais	TSR	7,2	Oct. 1991	65248
Measuring DSM Event Management Performance	M. Stockton	TSR	8,1	Spring 1992	65250
Network Statistics System	M. Miller	TSR	4,3	Oct. 1988	15748
Overview of DSM	P. Homan, B. Malizia, E. Reisner	TSR	4,3	Oct. 1988	15748
SCP and SCF: A General Purpose Implementation of the Subsystem Programmatic Interface	T. Lawson	TSR	4,3	Oct. 1988	15748
Tandem's Subsystem Programmatic Interface	G. Tom	TSR	4,3	Oct. 1988	15748
Using the Subsystem Programmatic Interface and Event Management Services	K. Stobie	TSR	4,3	Oct. 1988	15748
VIEWPOINT Operations Console Facility	R. Hansen, G. Stewart	TSR	4,3	Oct. 1988	15748
Writing Rules for Automated Operations	J. Collins	TSR	7,2	Oct. 1991	65248
<b>DYNAMITE</b>					
An Introduction to DYNAMITE Workstation Host Integration	S. Kosinski	TSR	1,2	June 1985	83935
The DYNAMITE Workstation: An Overview	G. Smith	TSR	1,2	June 1985	83935
<b>ENABLE</b>					
The ENABLE Program Generator for Multifile Applications	B. Chapman, J. Zimmerman	TSR	1,1	Feb. 1985	83934
<b>ENCOMPASS</b>					
The Relational Data Base Management Solution	G. Ow	TJ	2,1	Winter 1984	83931
<b>ENCORE</b>					
The ENCORE Stress Test Generator for On-line Transaction Processing Applications	S. Kosinski	TJ	2,1	Winter 1984	83931
<b>ENSCRIBE</b>					
Converting Database Files from ENSCRIBE to NonStop SQL	W. Weikel	TSR	6,1	March 1990	32986
<b>FASTSORT</b>					
FASTSORT: An External Sort Using Parallel Processing	J. Gray, M. Stewart, A. Tsukerman, S. Uren, B. Vaughan	TSR	2,3	Dec. 1986	83938
<b>FOX</b>					
Changes in FOX	N. Donde	TSR	1,2	June 1985	83935
Using FOX to Move a Fault-tolerant Application	C. Breighner	TSR	1,1	Feb. 1985	83934
<b>FUP</b>					
Online Reorganization of Key-Sequenced Tables and Files	G. Smith	TSR	6,2	Oct. 1990	46987

Article title	Author(s)	Publication	Volume, Issue	Publication date	Part number
<b>GUARDIAN 90</b>					
B00 Software Manuals	S. Olds	TSR	1,2	June 1985	83935
C00 Software Manuals	E. Levi	TSR	4,1	Feb. 1988	11078
Highlights of the B00 Software Release	K. Coughlin, R. Montevaldo	TSR	1,2	June 1985	83935
Improved Performance for BACKUP2 and RESTORE2	A. Khatri, M. McCline	TSR	1,2	June 1985	83935
Increased Code Space	A. Jordan	TSR	1,2	June 1985	83935
Managing System Time Under GUARDIAN 90	E. Nellen	TSR	2,1	Feb. 1986	83936
Message System Performance Enhancements	D. Kinkade	TSR	2,3	Dec. 1986	83938
Message System Performance Tests	S. Uren	TSR	2,3	Dec. 1986	83938
New GUARDIAN 90 Time-keeping Facilities	E. Nellen	TSR	1,2	June 1985	83935
New Process-timing Features	S. Sharma	TSR	1,2	June 1985	83935
NonStop II Memory Organization and Extended Addressing	D. Thomas	TJ	1,1	Fall 1983	83930
Overview of the C00 Release	L. Marks	TSR	4,1	Feb. 1988	11078
Robustness to Crash in a Distributed Data Base: A Nonshared-memory Multiprocessor Approach	A. Borr	TSR	1,2	June 1985	83935
Tandem's Approach to Fault Tolerance	B. Ball, W. Bartlett, S. Thompson	TSR	4,1	Feb. 1988	11078
The GUARDIAN Message System and How to Design for It	M. Chandra	TSR	1,1	Feb. 1985	83934
The Tandem Global Update Protocol	R. Carr	TSR	1,2	June 1985	83935
<b>INTEGRITY S2</b>					
Overview of the NonStop-UX Operating System for the Integrity S2	P. Norwood	TSR	7,1	April 1991	46988
<b>MEASURE</b>					
How to Set Up a Performance Data Base with MEASURE and ENFORM	M. King	TSR	2,3	Dec. 1986	83938
MEASURE: Tandem's New Performance Measurement Tool	D. Dennison	TSR	2,3	Dec. 1986	83938
<b>MULTILAN</b>					
Introduction to MULTILAN	A. Coyle	TSR	4,1	Feb. 1988	11078
Overview of the MULTILAN Server	A. Rowe	TSR	4,1	Feb. 1988	11078
Using the MULTILAN Application Interfaces	M. Berg, A. Rowe	TSR	4,1	Feb. 1988	11078
<b>NETBATCH-PLUS</b>					
NetBatch: Managing Batch Processing on Tandem Systems	D. Wakashige	TSR	5,1	April 1989	18662
NetBatch-Plus: Structuring the Batch Environment	G. Earle, D. Wakashige	TSR	6,1	March 1990	32986
<b>NONSTOP SQL</b>					
An Overview of NonStop SQL Release 2	M. Pong	TSR	6,2	Oct. 1990	46987
Concurrency Control Aspects of Transaction Design	W. Senf	TSR	6,1	March 1990	32986
Converting Database Files from ENSCRIBE to NonStop SQL	W. Weikel	TSR	6,1	March 1990	32986
Gateways to NonStop SQL	D. Slutz	TSR	6,2	Oct. 1990	46987
High-Performance SQL Through Low-Level System Integration	A. Borr	TSR	4,2	July 1988	13693
NonStop SQL Data Dictionary	R. Holbrook, D. Tsou	TSR	4,2	July 1988	13693

Article title	Author(s)	Publication	Volume, Issue	Publication date	Part number
<b>NONSTOP SQL (cont.)</b>					
NonStop SQL: The Single Database Solution	J. Cassidy, T. Kocher	TSR	5,2	Sept. 1989	28152
NonStop SQL Optimizer: Basic Concepts	M. Pong	TSR	4,2	July 1988	13693
NonStop SQL Optimizer: Query Optimization and User Influence	M. Pong	TSR	4,2	July 1988	13693
NonStop SQL Reliability	C. Fenner	TSR	4,2	July 1988	13693
Overview of NonStop SQL	H. Cohen	TSR	4,2	July 1988	13693
Parallelism in NonStop SQL Release 2	M. Moore, A. Sodhi	TSR	6,2	Oct. 1990	46987
Performance Benefits of Parallel Query Execution and Mixed Workload Support in NonStop SQL Release 2	S. Englert, J. Gray	TSR	6,2	Oct. 1990	46987
Tandem's NonStop SQL Benchmark	Tandem Performance Group	TSR	4,1	Feb. 1988	11078
The NonStop SQL Release 2 Benchmark	S. Englert, J. Gray, T. Kocher, P. Shah	TSR	6,2	Oct. 1990	46987
The Outer Join in NonStop SQL	J. Vaishnav	TSR	6,2	Oct. 1990	46987
<b>OSI</b>					
Building Open Systems Interconnection with OSI/AS and OSI/TS	R. Smith	TSR	6,1	March 1990	32986
The OSI Model: Overview, Status, and Current Issues	A. Dunn	TSR	5,1	April 1989	18662
<b>PATHFINDER</b>					
PATHFINDER—An Aid for Application Development	S. Benett	TJ	1,1	Fall 1983	83930
<b>PATHWAY</b>					
A New Design for the PATHWAY TCP	R. Wong	TJ	2,2	Spring 1984	83932
PATHWAY IDS: A Message-level Interface to Devices and Processes	M. Anderton M. Noonan	TSR	2,2	June 1986	83937
Pathway TCP Enhancements for Application Run-Time Support	R. Vannucci	TSR	7,1	April 1991	46988
The PATHWAY TCP: Performance and Tuning	J. Vatz	TSR	1,1	Feb. 1985	83934
Understanding PATHWAY Statistics	R. Wong	TJ	2,2	Spring 1984	83932
<b>PS MAIL</b>					
Enhancements to PS MAIL	R. Funk	TSR	3,1	March 1987	83939
<b>RDF</b>					
RDF: An Overview	J. Guerrero	TSR	7,2	Oct. 1991	65248
RDF Synchronization	F. Jongma, W. Senf	TSR	8,2	Summer 1992	69848
<b>SAFEGUARD</b>					
Dial-In Security Considerations	P. Grainger	TSR	7,2	Oct. 1991	65248
Distributed Protection with SAFEGUARD	T. Chou	TSR	2,2	June 1986	83937
Enhancing System Security With Safeguard	C. Gaydos	TSR	7,1	April 1991	46988
<b>SNAX</b>					
An Overview of SNAX/CDF	M. Turner	TSR	5,2	Sept. 1989	28152
A SNAX Passthrough Tutorial	D. Kirk	TJ	2,2	Spring 1984	83932
SNAX/APC: Tandem's New SNA Software for Distributed Processing	B. Grantham	TSR	3,1	March 1987	83939
SNAX/HLS: An Overview	S. Saltwick	TSR	1,2	June 1985	83935



Article title	Author(s)	Publication	Volume, Issue	Publication date	Part number
<b>SPOOLER</b>					
Sizing the Spooler Collector Data File	H. Norman	TSR	4,1	Feb. 1988	11078
<b>TACL</b>					
Debugging TACL Code	L. Palmer	TSR	4,2	July 1988	13693
TACL, Tandem's New Extensible Command Language	J. Campbell, R. Glascock	TSR	2,1	Feb. 1986	83936
<b>TAL</b>					
New TAL Features	C. Lu, J. Murayama	TSR	2,2	June 1986	83837
<b>TCM</b>					
Capacity Planning With TCM	W. Highleyman	TSR	7,2	Oct. 1991	65248
<b>TLAM</b>					
TLAM: A Connectivity Option for Expand	K. MacKenzie	TSR	7,1	April 1991	46988
<b>TMDS</b>					
C00 TMDS Performance	J. Mead	TSR	4,1	Feb. 1988	11078
Enhancements to TMDS	L. White	TSR	3,2	Aug. 1987	83940
Introducing TMDS, Tandem's New On-line Diagnostic System	J. Troisi	TSR	1,2	June 1985	83935
<b>TMF</b>					
Improvements in TMF	T. Lemberger	TSR	1,2	June 1985	83935
TMF and the Multi-Threaded Requester	T. Lemberger	TJ	1,1	Fall 1983	83930
TMF Autorollback: A New Recovery Feature	M. Pong	TSR	1,1	Feb. 1985	83934
<b>TNS/R</b>					
Debugging Accelerated Programs on TNS/R Systems	D. Cressler	TSR	8,1	Spring 1992	65250
Improving Performance on TNS/R Systems With the Accelerator	M. Blanchet	TSR	8,1	Spring 1992	65250
Overview of Tandem NonStop Series/RISC Systems	L. Faby, R. Mateosian	TSR	8,1	Spring 1992	65250
<b>TRANSFER</b>					
The TRANSFER Delivery System for Distributed Applications	S. Van Pelt	TJ	2,2	Spring 1984	83932
<b>TXP</b>					
The High-Performance NonStop TXP Processor	W. Bartlett, T. Houy, TJ D. Meyer		2,1	Winter 1984	83931
The NonStop TXP Processor: A Powerful Design for On-line Transaction Processing	P. Oleinick	TJ	2,3	Summer 1984	83933
<b>V8</b>					
The V8 Disc Storage Facility: Setting a New Standard for On-line Disc Storage	M. Whiteman	TSR	1,2	June 1985	83935
<b>VIEWSYS</b>					
VIEWSYS: An On-line System-resource Monitor	D. Montgomery	TSR	1,2	June 1985	83935
<b>VLX</b>					
NonStop VLX Hardware Design	M. Brown	TSR	2,3	Dec. 1986	83938
NonStop VLX Performance	J. Enright	TSR	2,3	Dec. 1986	83938
The VLX: A Design for Serviceability	J. Allen, R. Boyle	TSR	3,1	March 1987	83939
<b>XL8</b>					
Data-encoding Technology Used in the XL8 Storage Facility	D.S. Ng	TSR	2,2	June 1986	83937
Plated Media Technology Used in the XL8 Storage Facility	D.S. Ng	TSR	2,2	June 1986	83937

Article title	Author(s)	Publication	Volume, Issue	Publication date	Part number
<b>MISCELLANEOUS<sup>1</sup></b>					
A Performance Retrospective	P. Oleinick	TSR	2,3	Dec. 1986	83938
Batch Processing in Online Enterprise Computing	T. Keefauver	TSR	6,2	Oct. 1990	46987
Buffering for Better Application Performance	R. Mattran	TSR	2,1	Feb. 1986	83936
Capacity Planning Concepts	R. Evans	TSR	2,3	Dec. 1986	83938
Configuring Tandem Disk Subsystems	S. Sittler	TSR	2,3	Dec. 1986	83938
Connecting Terminals and Workstations to Guardian 90 Systems	E. Siegel	TSR	8,2	Summer 1992	69848
Credit-authorization Benchmark for High Performance and Linear Growth	T. Chmiel, T. Houy	TSR	2,1	Feb. 1986	83936
Data-window Phase-margin Analysis	A. Painter, H. Pham, H. Thomas	TSR	2,2	June 1986	83937
Estimating Host Response Time in a Tandem System	H. Horwitz	TSR	4,3	Oct. 1988	15748
Getting Optimum Performance from Tandem Tape Systems	A. Khatri	TSR	2,3	Dec. 1986	83938
Network Design Considerations	J. Evjen	TSR	5,2	Sept. 1989	28152
New Software Courses	M. Janow	TSR	1,2	June 1985	83935
New Software Courses	J. Limper	TSR	4,1	Feb. 1988	11078
Optimizing Batch Performance	T. Keefauver	TSR	5,2	Sept. 1989	28152
Optimizing Sequential Processing on the Tandem System	R. Welsh	TJ	2,3	Summer 1984	83933
Performance Considerations for Application Processes	R. Glasstone	TSR	2,3	Dec. 1986	83938
Performance Measurements of an ATM Network Application	N. Cabell, D. Mackie	TSR	2,3	Dec. 1986	83938
Peripheral Device Interfaces	J. Blakkan	TSR	3,2	Aug. 1987	83940
Predicting Response Time in On-line Transaction Processing Systems	A. Khatri	TSR	2,2	June 1986	83937
Remote Support Strategy	J. Eddy	TSR	3,1	March 1987	83939
Streaming Tape Drives	J. Blakkan	TSR	3,2	Aug. 1987	83940
Subscription Policy for Software Manuals	T. McSweeney	TSR	2,1	Feb. 1986	83936
Tandem's New Products	C. Robinson	TSR	2,1	Feb. 1986	83936
Tandem's New Products	C. Robinson	TSR	2,2	June 1986	83937
Tandem's Software Support Plan	R. Baker, D. McEvoy	TSR	3,1	March 1987	83939
Terminal Connection Alternatives for Tandem Systems	J. Simonds	TSR	5,1	April 1989	18662
Terminal Selection	E. Siegel	TSR	8,2	Summer 1992	69848
The Performance Characteristics of Tandem NonStop Systems	J. Day	TJ	1,1	Fall 1983	83930
The Role of Optical Storage in Information Processing	L. Sabaroff	TSR	3,2	Aug. 1987	83940

<sup>1</sup>This category is composed of articles that contain product information but are not specifically product-related.

# Tandem Systems Review Order Form

The *Tandem Systems Review* is published quarterly. Subscriptions are \$75 per year; single copies are \$20 each.

I am a current Tandem customer. Please invoice me for the requests on this form.

My Tandem sales representative is \_\_\_\_\_

I am enclosing a check or money order for the requests on this form. (Make check payable to Tandem Computers Incorporated.)

## Subscription Information

- New subscription  
 Update to subscription information  
Subscription number: \_\_\_\_\_

*Your subscription number is in the upper right corner of the mailing label.*

COMPANY \_\_\_\_\_

NAME \_\_\_\_\_

JOB TITLE \_\_\_\_\_

DIVISION \_\_\_\_\_

ADDRESS \_\_\_\_\_

COUNTRY \_\_\_\_\_

TELEPHONE NUMBER (include all codes for U.S. dialing) \_\_\_\_\_

Title or position:

- President/CEO  
 Director/VP information services  
 MIS/DP manager  
 Software development manager  
 Programmer/analyst  
 System operator  
 End user  
 Other: \_\_\_\_\_

Your association with Tandem:

- Tandem customer  
 Third-party vendor  
 Consultant  
 Other: \_\_\_\_\_

## Back Issue Requests

Number of copies

### *Tandem Systems Review*

- |                                 |                                  |
|---------------------------------|----------------------------------|
| _____ Vol. 1, No. 1, Feb. 1985  | _____ Vol. 6, No. 1, March 1990  |
| _____ Vol. 1, No. 2, June 1985  | _____ Vol. 6, No. 2, Oct. 1990   |
| _____ Vol. 2, No. 1, Feb. 1986  | _____ Vol. 7, No. 1, April 1991  |
| _____ Vol. 2, No. 2, June 1986  | _____ Vol. 7, No. 2, Oct. 1991   |
| _____ Vol. 2, No. 3, Dec. 1986  | _____ Vol. 8, No. 1, Spring 1992 |
| _____ Vol. 3, No. 1, March 1987 | _____ Vol. 8, No. 2, Summer 1992 |
| _____ Vol. 3, No. 2, Aug. 1987  |                                  |
| _____ Vol. 4, No. 1, Feb. 1988  |                                  |
| _____ Vol. 4, No. 2, July 1988  |                                  |
| _____ Vol. 4, No. 3, Oct. 1988  |                                  |
| _____ Vol. 5, No. 1, April 1989 |                                  |
| _____ Vol. 5, No. 2, Sept. 1989 |                                  |

### *Tandem Journal*

- |                                  |                                  |
|----------------------------------|----------------------------------|
| _____ Vol. 1, No. 1, Fall 1983   | _____ Vol. 2, No. 2, Spring 1984 |
| _____ Vol. 2, No. 1, Winter 1984 | _____ Vol. 2, No. 3, Summer 1984 |

For more information or to place your order by telephone, call: 1-800-473-5868

Send this form to:  
Tandem Computers Incorporated  
*Tandem Systems Review*, Loc 216-05  
18922 Forge Drive  
Cupertino, CA 95014-0701

Tandem employees must order their subscriptions and back issues through Courier.

Menu sequence: Marketing Information → Literature Orders → Technical Marketing Pubs → Tandem Systems Review

▲ FOLD



▲ FOLD

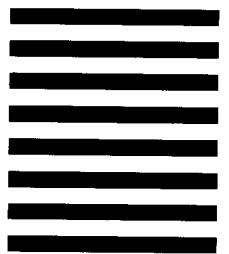
**BUSINESS REPLY MAIL**

FIRST CLASS PERMIT NO. 482 CUPERTINO, CA U.S.A.

POSTAGE WILL BE PAID BY ADDRESSEE

**TANDEM SYSTEM REVIEW**  
LOC 216-05  
TANDEM COMPUTERS INCORPORATED  
19333 VALLCO PARKWAY  
CUPERTINO, CA 95015-9862

NO POSTAGE  
NECESSARY  
IF MAILED  
IN THE  
UNITED STATES



▼ FOLD

▼ FOLD

# TandemSystemsReviewReaderSurvey

The purpose of this questionnaire is to help the *Tandem Systems Review* staff select topics for publication. Postage is prepaid when mailed in the United States. Readers outside the U.S. should send their replies to their nearest Tandem sales office.

1. How useful is each article in this issue?

- |   |                                  |                                      |  |
|---|----------------------------------|--------------------------------------|--|
| <i>Product Update</i>   |                                  |                                      |  |
| 01 <input type="checkbox"/> Indispensable                           | 02 <input type="checkbox"/> Very | 03 <input type="checkbox"/> Somewhat | 04 <input type="checkbox"/> Not at all |
| <i>RDF Synchronization</i>  |                                  |                                      |  |
| 05 <input type="checkbox"/> Indispensable                           | 06 <input type="checkbox"/> Very | 07 <input type="checkbox"/> Somewhat | 08 <input type="checkbox"/> Not at all |
| <i>Terminal Selection</i>   |                                  |                                      |  |
| 09 <input type="checkbox"/> Indispensable                           | 10 <input type="checkbox"/> Very | 11 <input type="checkbox"/> Somewhat | 12 <input type="checkbox"/> Not at all |
| <i>Connecting Terminals and Workstations to Guardian 90 Systems</i> |                                  |                                      |  |
| 13 <input type="checkbox"/> Indispensable                           | 14 <input type="checkbox"/> Very | 15 <input type="checkbox"/> Somewhat | 16 <input type="checkbox"/> Not at all |
| <i>Technical Information and Education</i>                          |                                  |                                      |  |
| 17 <input type="checkbox"/> Indispensable                           | 18 <input type="checkbox"/> Very | 19 <input type="checkbox"/> Somewhat | 20 <input type="checkbox"/> Not at all |
| <i>Ongoing Support: InfoWay Online Support Service</i>              |                                  |                                      |  |
| 21 <input type="checkbox"/> Indispensable                           | 22 <input type="checkbox"/> Very | 23 <input type="checkbox"/> Somewhat | 24 <input type="checkbox"/> Not at all |

2. I specifically would like to see more articles on (select one):

- |   |   |
|---|---|
| 25 <input type="checkbox"/> Overview discussions of new products and enhancements.  | 26 <input type="checkbox"/> Performance and tuning information.       |
| 27 <input type="checkbox"/> High-level overviews on Tandem's approach to solutions. | 28 <input type="checkbox"/> Application design and customer profiles. |
| 29 <input type="checkbox"/> Technical discussions of product internals.             |   |
| 30 <input type="checkbox"/> Other _____   |   |

3. Your title or position:

- |   |  |   |
|---|--|---|
| 31 <input type="checkbox"/> President, VP, Director | 32 <input type="checkbox"/> Systems analyst    | 33 <input type="checkbox"/> System operator |
| 34 <input type="checkbox"/> MIS manager             | 35 <input type="checkbox"/> Software developer | 36 <input type="checkbox"/> End user        |
| 37 <input type="checkbox"/> Other _____             |  |   |

4. Your association with Tandem:

- |   |   |  |  |
|---|---|--|--|
| 38 <input type="checkbox"/> Tandem customer | 39 <input type="checkbox"/> Tandem employee | 40 <input type="checkbox"/> Third-party vendor | 41 <input type="checkbox"/> Consultant |
| 42 <input type="checkbox"/> Other _____     |   |  |  |

5. Comments

---



---



---

NAME \_\_\_\_\_

COMPANY NAME \_\_\_\_\_

ADDRESS \_\_\_\_\_

▲ FOLD



▲ FOLD

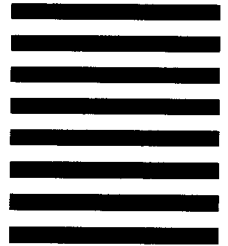
**BUSINESS REPLY MAIL**

FIRST CLASS      PERMIT NO. 482      CUPERTINO, CA U.S.A.

POSTAGE WILL BE PAID BY ADDRESSEE

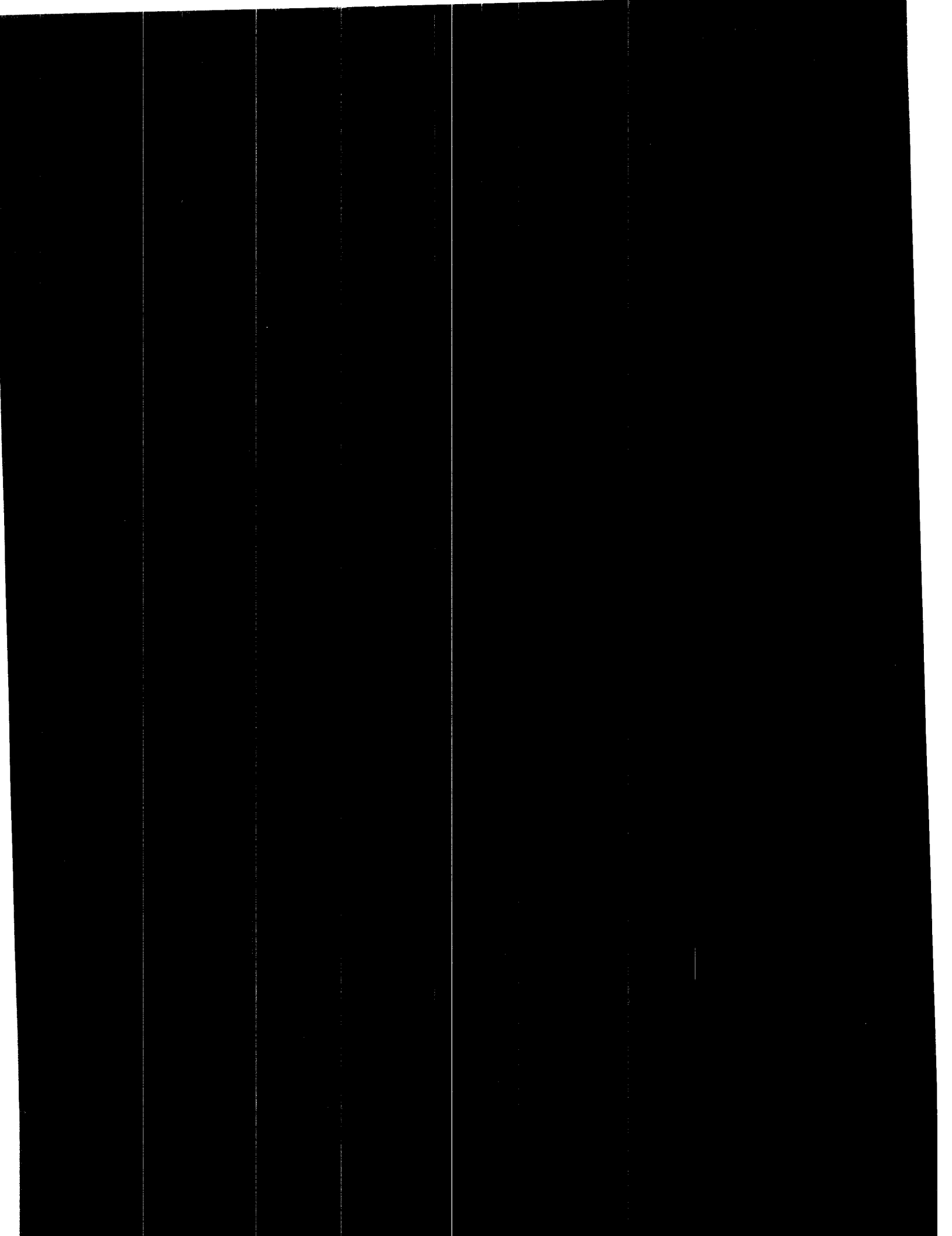
**TANDEM SYSTEM REVIEW**  
LOC 216-05  
TANDEM COMPUTERS INCORPORATED  
1933 VALLCO PARKWAY  
CUPERTINO, CA 95015-9862

NO POSTAGE  
NECESSARY  
IF MAILED  
IN THE  
UNITED STATES



▼ FOLD

▼ FOLD





Tandem Computers Incorporated  
19333 Vallco Parkway  
Cupertino, CA 95014-2599

MARC BRANDIFINO  
LOC NUM 56-00  
NEW YORK NY DOWNTOWN DISTRICT