

User's Guide

Technetron XPO

EXtended Processing Option

Trademarks

The following are trademarks of International Business Machines Corporation.

MVS/SP
System/370
S/370

Registered Trademarks

IBM is a registered trademark of International Business Machines Corporation.

CA-IDMS is a registered trademark of Computer Associates International, Inc.

IDMS is used interchangeably with CA-IDMS in this document.

PREFACE

Technetron has continued its enhancement of CA-IDMS related software. The latest enhancement enables the DBMS to operate continuously, and is IDMS release 14.0 compatible. The enhancement is built on the proven operation of Technetron Duplex Subsystem (TDS) with additional features integrated with Technetron Dynamic Facility (TDF). The two components, TDS and TDF, will be collectively referred to as the EXTended Processing Option, or XPO for short.

Technetron has made additional effort to provide and support the same software for the two later releases of CA-IDMS. This should simplify the migration process from CA-IDMS 12 to CA-IDMS 14. Technetron software installation/integration procedure has not been altered. Users familiar with Technetron software operation will only need to download the software from the new release tape and apply the same site dependent procedure to resume operation.

TDS is made of Technetron's four duplex products. The duplex software maintains an extra set of IDMS external entities. The four TDS products are listed by their name, release information and compatibility:

<u>PRODUCT NAME</u>	<u>LEVEL</u>	<u>RELEASE INFORMATION</u>	
		<u>DATE</u>	<u>COMPATIBILITY</u>
ARCHIVE/DUPLEX	Rel 4.0	12/11	CA-IDMS 12.x, 14.x
DB/DUPLEX	Rel 7.0	12/11	CA-IDMS 12.x, 14.x
DUPLEX/SYNCHRO	Rel 3.0	12/11	CA-IDMS 12.x, 14.x
JOURNAL/DUPLEX	Rel 7.0	12/11	CA-IDMS 12.x, 14.x

There were significant enhancements made to the duplex software including the DUPLEX/SYNCHRO online component, which operates with its own external entity, the duplex queue, is an essential part of duplex operation allowing CA-IDMS to operate continuously. The key in operating IDMS 24 hours a day is to provide control over duplex processing and proper maintenance of the duplex entities as if they were the primary entities, while normal IDMS operates with the real primary parts.

The control mechanism is provided by TDF. While nonstop operation is important, the DBMS must preserve its database integrity first. DB/DYNAM has been thoroughly enhanced with many new features including starting and stopping of the duplex queuing process. The two products of TDF are listed below:

<u>PRODUCT NAME</u>	<u>LEVEL</u>	<u>RELEASE INFORMATION</u>	
		<u>DATE</u>	<u>COMPATIBILITY</u>
DB/DYNAM	Rel 7.8	12/11	CA-IDMS 12.x, 14.x
DB/INFO	Rel 6.7	12/11	CA-IDMS 12.x, 14.x

Integrating the six products together with IDMS has made XPO the true 24X7 solution for IDMS. This document explains the operation and use of all Technetron software products at the current release level. This manual should be used as the reference for all Technetron installation regardless of Technetron product configurations. All individual duplex product User's Guides, previous TDS User's Guides, DB/DYNAM User's guide are superseded by this edition of the XPO User's Guide.

DOCUMENT OUTLINE

- Section 1: Provides an overview of mainframe database applications and the importance of their availability. XPO is introduced with general description of all TDS and TDF products, and how the three different operation modes of XPO enable CA-IDMS to operate continuously.
- Section 2: Describes the tasks involved in operating TDS and TDF. The section includes discussion on customizing Technetron software, and the integration process with existing IDMS systems; topics include duplex table generation, duplex queue operation, dynamic facility capability including online duplex operation alteration and duplex status queries.
- Section 3: Details some of the recovery issues for online, local mode and utility operations which may occurred when operating with duplex software. Generalized recovery procedures are provided as guidelines under different operating modes.
- Section 4: Lists messages and codes produced by the system.
- Section 5: Describe the installation procedure for Technetron Software Licensed Products.

RELATED DOCUMENTS

DUPLEXING AND EXTENDED PROCESSING - AN OVERVIEW

CA-IDMS System Software - System Operations

TABLE OF CONTENTS

SECTION 1: INTRODUCTION	1 - 1
Information Systems	1 - 1
System Availability	1 - 1
Database and Journal	1 - 2
Duplexing and Mirroring	1 - 3
XPO - TDS Overview	1 - 4
XPO - TDF Overview	1 - 7
SECTION 2: SYSTEM OPERATION	2 - 1
XPO - 24x7 Enhancement	2 - 1
Duplex Queue	2 - 4
Duplex Table	2 - 5
TDS Batch Component	2 - 10
File Preparation and Usage	2 - 12
System Integration	2 - 13
DB/DYNAM	2 - 16
ALLOCATE Command	2 - 18
DEALLOCATE Command	2 - 20
CLOSE Command	2 - 22
SWAP Command	2 - 23
DUPLEX Command	2 - 24
BYPASS Command	2 - 24
START Command	2 - 25
DB/INFO	2 - 25
SECTION 3: RECOVERY OPERATION	3 - 1
Central Version Recovery	3 - 1
CV Database Recovery Procedure	3 - 2
CV Journal Recovery Procedure	3 - 3
Local Mode Recovery	3 - 5
Local Mode Database Recovery Procedure	3 - 6
Local Mode Journal Recovery Procedure	3 - 8
Utility Recovery	3 - 9
IDMSAJNL Journal Recovery Procedure	3 - 10
IDMSAJNL Archive Recovery Procedure	3 - 11

TABLE OF CONTENTS

SECTION 4: MESSAGES AND CODES	4 - 1
Messages	4 - 1
Generated by DB/DYNAM.....	4 - 1
Generated by DB/INFO	4 - 17
Generated by Technetron Duplex Subsystem.....	4 - 18
Codes.....	4 - 20
Generated by Technetron Dynamic Facility.....	4 - 20
Generated by Technetron Duplex Subsystem.....	4 - 21
SECTION 5: INSTALLATION	5 - 1
Tape and Content.....	5 - 1
Step 1 - Unload Tape	5 - 3
Step 2 - Modify and Execute.....	5 - 4
INSTDPLX	5 - 4
INSTDYNM	5 - 5
CUSTDYNM	5 - 6
FIGURE 1 - Technetron Duplex Subsystem.....	2 - 1
FIGURE 2 - Enhanced TDS Operation	2 - 2
FIGURE 3 - Duplex Queuing	2 - 3
FIGURE 4 - Duplex Synchronization.....	2 - 3

CONVENTIONS

The conventions below are used in this manual to define syntax and JCL.

SYNTAX

<u>Notation</u>	<u>Meaning</u>
UPPERCASE CHARACTERS	Specify exactly as shown.
lowercase characters	Variable field, replace this with a user-supplied value.
[]	Specify none or one of the options enclosed by the brackets.
{ }	Specify one of the options enclosed by the braces.
←	The corresponding value is the default.

JCL

<u>Notation</u>	<u>Meaning</u>
UPPERCASE CHARACTERS	Specify exactly as shown.
lowercase characters	Replace with the appropriate installation value.

INTRODUCTION

The introductory section previews Technetron EXTended Processing Option (XPO) by describing information systems (IS) in general, and the need to maximize the most desirable system attribute; availability. It is followed by the discussion on the subject of system availability protection in the mainframe database environment.

The first part of Section 1 continues with the discussion of DataBase Management Systems (DBMS) and their external entities. Focusing largely on the mainframe DBMS and its journaling capability, the concept of dual journaling is introduced with emphasis on duplexing and mirroring. The discussion of duplexing is expanded to include database mirroring which leads to the summary description of Technetron's implementation for duplexing of all IDMS external entities.

The second part of the first section introduces the two major components of XPO, Technetron Duplex Subsystem (TDS) and Technetron Dynamic Facility (TDF), with a brief description of all their products. This section also includes the discussion on the advantages of maintaining a duplexed IDMS environment, and how TDS and its operation form the foundation for continuous IDMS operation.

INFORMATION SYSTEMS

In today's business world, information is needed for all aspects of most companies operations. Information systems have become more than just a vital part of those companies' ability to operate. Today, information systems are often being considered or measured as the competitive edge that one company may hold over another.

Database information systems have become an integral part of most large firm's operation. It is difficult to image any sizable corporation's automated processing is without the application of database systems. Since a database is the repository for information as well as the raw data to be efficiently processed into information, it is often used as the basic building block of a sophisticated information system.

Knowing a company's operation and its competitiveness may depend on the performance of its automated systems, increasing the system functionality and availability would strengthen the company's business position. It is therefore beneficial for most companies to address the fundamentals of their information systems including the database and all of its associated entities.

SYSTEM AVAILABILITY

There are many issues to be concerned with when enhancing an information system. Certain system trade-offs, such as performance vs. security and availability vs. serviceability, are often made as a part of the enhancement process. Protection is

SECTION 1: INTRODUCTION

DATABASE AND JOURNAL

one area of system enhancement that relates to two different system attributes, availability and security.

The most important attribute of a system is its availability. An user-friendly and fully functional system with strict access control is rendered useless if it is not available. So, availability should be the first system attribute of consideration when addressing the protection issue.

In the past, the emphasis on protection in the IBM mainframe database environment has been that of security. The availability protection on the database systems has been largely ignored. Sensitive, critical and time-dependent data, and the applications that utilize them, have up till now not been accorded with the availability protection that is proportional to their importance in a company.

More recently, the growing need and the demand for continual access to information by international financial institutions and manufacturers have brought out the importance of information systems and their availability. Even as corporations offload their processing onto more cost conservative platforms such as personal workstations and midrange computers, greater emphasis is being placed on the mainframe application availability.

For mainframe database systems which are often the central point of the process that other systems depend on, their availability should be protected to the fullest extent. The availability of a mainframe database application is closely related to the DBMS on which it was implemented.

Mainframe DBMSs have mainly addressed its availability issue through the operating environment which they exist. The specific issues of availability extension will become more evident by reviewing the mainframe DBMS in general and the common implementation of its recovery mechanism.

DATABASE and JOURNAL

A DBMS is the set of software and the data structure which it support that allows efficient access and manipulation of data. Having the extra layer of software effectively manage the resource between the operating system and the application sets the foundation for shared data. Although DBMSs started out as a way to reduce data redundancy, they have evolved to become an essential part of implementing complex IS offering efficient and effective methods of processing information.

Mainframe DBMSs provides their users the capability to process large volume of data and high number of transactions by maintaining the data on secondary direct access devices. The external entities are referred to as the database for which the DBMS will ensure the information is properly kept and be recoverable in the event of a system failure.

A DBMS guards against general system failures by recording the changes that have been made to the database to another secondary storage device called the journal. So, after a system failure, a database can be recreated by restoring a prior copy of the database and applying the journal content to the copy. However, there is one type of system failures, I/O error, which may occur on either the database or the journal which can extensively delay recovering a database system.

Since journals are used to preserve a database's integrity after failures, and with the possibility of having problems with them, the best way to ensure their availability is to have two sets of them. The process of maintaining two sets of journal is referred to as dual journaling. The best implementation of a dual journal environment would be to ensure the interchangeability of the two journals, thereby enabling the DBMS to utilize the same recovery mechanism with either journal.

DUPLEXING and MIRRORING

Journal duplexing is a type of dual journaling whereby two sets of one journal is constructed simultaneously. The process of keeping the two journals as an exact duplicate of each other is referred to as journal mirroring. It is the preferred method of dual journaling due to the extra assurance of the availability of the critical recovery resource with the same recovery procedure. The only issue which surrounds journal mirroring is the performance impact it may have on the applications.

For the mainframe network DBMS, CA-IDMS, is well protected with its sophisticated journaling methodology and automatic restart capability. However, IDMS is also one of the DBMSs that does not mirror its journal. It depends on only one set of journal for recovery. Although CA-IDMS recovery can also be performed with archives, that process is usually more time consuming and has a higher level of exposure to risks.

Technetron Duplex Subsystem complements CA-IDMS with its journal mirroring facility, JOURNAL/DUPLEX. It furthers the duplex concept by allowing CA-IDMS users to duplicate a portion of or an entire database under their control using the database mirroring product, DB/DUPLEX, all without impacting system performance. Duplexing the database offers not only the availability assurance but also provides the foundation for continuous processing.

In CA-IDMS recovery situations involving I/O errors without duplex protection, critical data have been treated to the same manual recovery delays as have relatively non-critical data. Indeed, it has never been said that the normal manual recovery approach to I/O error recovery will restore the data to its exact condition just prior to the error, since the journal may be unreadable. Imagine an international electronic funds transfer operation that could not reconstruct the transactions that occurred just prior to an I/O error. Besides the lost time, the possibility of misplacing millions of dollars electronically is more than likely!

Technetron Duplex Subsystem can easily help its users avoid the above situation by duplicating the critical resource, journals and vital database files. The other benefits

SECTION 1: INTRODUCTION

XPO – TDS OVERVIEW

that TDS provides, such as extending the online processing time frame and balancing the load of the system, are the added advantages of the software.

XPO - TDS OVERVIEW

XPO refers to Technetron's duplex software and the dynamic configuration facility. The duplex software component, TDS, duplexes CA-IDMS external entities to form the foundation for extending the DBMS's availability. The dynamic software, referred to as TDF, provides the capability to control the duplex operation dynamically. The non-stop processing environment is supported by integrating the two components with CA-IDMS. The rest of this section provides an overview of TDS and TDF.

There are two types of secondary device media that TDS replicates. For the more reliable media, direct access device, only one copy is made. For the removable type of media, tape or cartridge, up to 9 copies can be produced. We refer to the part of TDS which handles the direct access devices as the online component, and the part that handles the portable media as the batch component.

Although the major duplex components are functionally integrated with CA-IDMS, the actual duplication process is segregated, and the duplex entities created are independent of CA-IDMS. TDS mirrors the entire set of CA-IDMS external entities, and it interfaces directly with the operating system to maintain the duplex set of those entities.

TDS contains four software products. Three out of the four duplex products perform their specific function independently of the others and therefore can operate as individual add-ons to CA-IDMS. The products have been designed and developed to ensure their collective performance is not compromised by the modularize approach. The TDS modules are strategically placed, and are programmed to maximize existing resource utilization while incurring a minimum amount of overhead.

TDS provides the extra level of protection by duplicating CA-IDMS entities according to the need of its users. The three products that perform the actual duplication are:

- **DB/DUPLEX** - Operates as a standard exit off the IDMS-DB physical input/output routine, IDMSDBIO. It mirrors IDMS Basic Direct Access Method (BDAM) database files. DB/DUPLEX shares the same duplex driver module with JOURNAL/DUPLEX's direct access (disk) journal mirroring facility.
- **ARCHIVE/DUPLEX** - Operates as a standard exit off the CA-IDMS journal archive utility, IDMSAJNL. It can simultaneously mirror up to 9 copies of the primary journal archive.
- **JOURNAL/DUPLEX** - For disk journal, it operates off the same exit as DB/DUPLEX. It can be used as a stand-alone product or as an add-on to

DB/DUPLEX to duplicate the disk journal in parallel with the primary. It operates off another IDMS-DB user exit to produce up to 9 copies of the primary sequential journal file.

The fourth product, **DUPLEX/SYNCHRO**, is the product that synchronizes the primary database and the disk journal with their duplex counterpart. It has two modes of operation, batch and online. When DUPLEX/SYNCHRO is active in the online mode, it does the queuing and the synchronization of the duplex queue enabling XPO to go from one processing state to another. The batch mode processing will copy database or disk journal files in sets of 8, or it will synchronize a duplex database using the duplex queue.

There are no special modifications made to CA-IDMS or within MVS to accommodate TDS. The online duplex processing is controlled by its predefined duplex table, DPLXTABL, and by the specification of Job Control Language (JCL) or with DB/DYNAM.

TDS provides many advantages directly and indirectly. The discussion of advantages in the present form is meant to assist users in reviewing the CA-IDMS environment (simplex) without TDS and compare it to a duplex environment; it should also help them in analyzing their options, derive new or change existing procedures in the duplex environment. A fully duplexed environment holds the major advantages of:

- Quicker and easier recovery - A database that has suffered an I/O error, and that is protected by TDS, can be quickly recovered by changing the startup JCL of Central Version (CV), or the local mode application JCL, to point the database's DDname to the duplicate database. Now recovery can continue as for the case of a power failure which, in the case of CV, means restarting CV. This is much faster and simpler than the conventional manual I/O error recovery of unload/JFIX/restore/rollforward/warmstart/ PFIX for forward recovery or unload/JFIX/rollback/warmstart for backward recovery.
- More complete data recovery - A journal that has suffered an I/O error, and that is protected by TDS, can be recovered by changing the startup JCL of CV, or the local mode JCL of an application job, to point the journal's DDname to the duplicate journal. CV can then be warmstarted and recovery is complete, or for local mode jobs, the same installation procedure for local mode recovery after a power failure can be invoked. As well as providing quicker recovery than normal procedures, the data on the duplicate journal can be read, so no database updates have been lost, no database files need to be restored and rolled forward and, for CV, no journals need to be offloaded and JFIX'ed.

In the case of archive journal, when an I/O error occurs on either the primary or the duplex, JCL can be easily changed to point the archive journal's DDname to the duplicate archive to complete the journal offload process. The successful completion of archive journal assures the manual recovery

SECTION 1: INTRODUCTION

XPO – TDS OVERVIEW

can continue and be completed without lost of data when it is needed. As well as providing quicker recovery than normal procedures, the data on the duplicate archive can be read, so no archived database updates have been lost.

- More timely off-site storage procedure - ARCHIVE/DUPLEX and JOURNAL/DUPLEX can be used to create one or more duplicate sequential journals, and one or more of these can be sent off-site while retaining the primary sequential journal on-site to provide timely recovery.
- More comprehensive data backup - It is possible to delay backup or backup less frequently on large or multi volume databases with full duplex operation. Since the duplex files are mirror images of the primary files, they can be viewed as the primary files backup. In reality, either the primary or the duplex files may serve as a backup copy of the other, thus providing TDS users with greater flexibility and a higher level of comfort in any CA-IDMS operation. The full duplex user can backup one set of database while processing the other, or simply let one database acts as the backup, complete a whole processing cycle, and then (re)synchronize the two databases.

The creation of one or more duplicate copies of the sequential journal at the time that the primary sequential journal is written is a safeguard against any I/O error on the primary journal. For example, some database administrators (DBAs) run the IBM utility IEBGENER to create a copy of the journal archive after IDMSAJNL has been run. But this strategy ignores the possibility of an I/O error while reading the primary archive journal. In this case, the primary archive is unreadable, the copy has not been created, and the disk journal has been condensed so all the backup data is lost!

The above advantages have highlighted the uniqueness of TDS, allowing users to maintain control over the enhanced environment. This capability has allowed users to parallel process in the duplex environment; it is a major factor for TDS users to successfully extend their CA-IDMS operation. Functionally, proper integration of duplex entities in the normal processing means more than just insurance; it has translated into simplified recovery operation and extended normal processing for the TDS users.

XPO - TDF OVERVIEW

TDF is the dynamic configuration and query facility of XPO. It provides XPO users the capability to alter duplex operation and query system information online. TDF is comprised of two Technetron software product, DB/DYNAM and DB/INFO. Their function summary is as follow:

- **DB/DYNAM** - is a set of IDMS-DC modules providing flexibility for CV operations. DB/DYNAM enables its users to dynamically configure a CV's operating environment including swapping of database files and alteration of duplex operation. Its main function is to provide users with operational control so a CV installed with DB/DYNAM would be able to operate continuously or near nonstop.
- **DB/INFO** - is a set of IDMS-DC modules providing information about CV operations. DB/INFO lists the datasets for load library, database, journal, and duplex files. It also shows the duplex table (DPLXTABL) definition and returns the status of the duplex processing online.

In general, TDF offers its users the following options:

- 24-hour processing - When integrated with TDS, TDF provides the control mechanism in operating a CV continuously. Even operating as a stand-alone utility, DB/DYNAM enables its users to widen their batch window and extend the CV online operation by providing more control over the system on the file level instead of the whole database.
- System error correction - TDF's dynamic allocation and deallocation capability allows its users to exchange a damaged file without bringing down the entire system.

TDF is thoroughly integrated with IDMS and TDS. CA-IDMS users without TDS will gain more control over their operating environment using TDF while XPO users will be able to operate IDMS continuously using TDS as the foundation and TDF as the control.

XPO – 24X7 ENHANCEMENT

XPO was implemented based on the principle of concurrent processing with one dynamic process and one static process. Building on TDS maintenance of the extra set of IDMS external entities, the two processes, CV processing and backup processing, will utilize the two copies of one database.

The dynamic process, CV, alters the content of the database, and is the objective of the 24X7 enhancement to preserve its operation with little or no interruption. The static process, backup, is the shorter of the two processes, but it requires the database content to remain unchanged while it is active. The two processes conflict with each other in their operational conflict by providing two identical copies of the same database would reduce the problem to different database content after an instance of concurrent processing.

The enhancement is to ensure the proper mapping of the database copy to the process, and resynchronizes the two copies of the database when the backup processing is complete. Since TDS maintains the mirrored (duplex) copy of IDMS database, the duplex database will be utilized by the backup, and made to be the mirrored copy of the primary database again after the backup is finished.

The 24X7 enhancement was designed to utilize TDS as the platform to build the nonstop operating environment. The independent duplex operation in maintaining its own set of IDMS external entities makes the enhancement straightforward. The normal CV processing with TDS, depicted below, shows TDS's online component, JOURNAL/DUPLEX and DB/DUPLEX, having direct control over their entities.

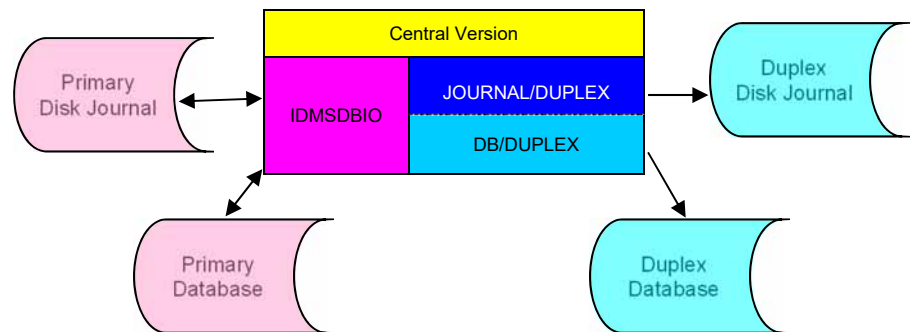


Figure 1 – Technetron Duplex Subsystem

SECTION 2: SYSTEM OPERATION
XPO – 24X7 ENHANCEMENT

The 24X7 enhancement integrates an additional DUPLEX/SYNCHRO module with the TDS Online Component. Although the new module is not active during the normal process of CV operation, it requires a pre-allocated set of two disk files. The files are referred to as the Duplex Queue, it is used to queue the database changes when the duplex database is taken offline for preventive maintenance.

Figure 2 illustrates what a CV enhanced with the 24X7 option would look like during its normal processing. Note the line between the duplex queue and DUPLEX/SYNCHRO represents the allocation of queue files, the real connection, open duplex files, is not established until the start of the duplex queuing process. There is no performance impact at this point, TDS operation remains the same.

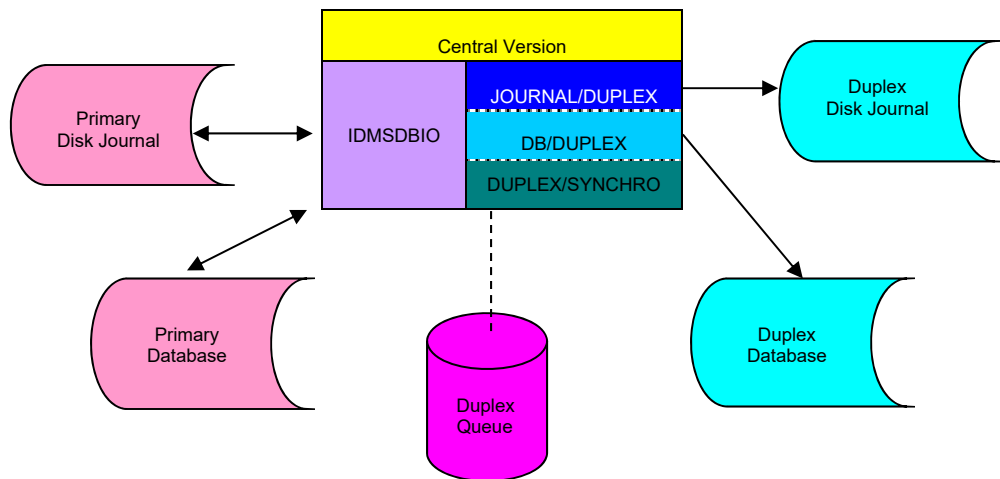


Figure 2 – Enhanced TDS Operation

The 24X7 enhancement transforms TDS-CV processing into three states; normal, queuing and synchronization. Each state processes with only duplex entities on the system level; their independent processing from IDMS-CV ensures normal CV operation is totally preserved.

The queuing process starts when the duplex database is taken offline for backup. Figure 3 depicts the swap of connection between the duplex database and the duplex queue. Since the duplex database is not being updated at this time, it can be utilized for the backup processing.

SECTION 2: SYSTEM OPERATION XPO – 24X7 ENHANCEMENT

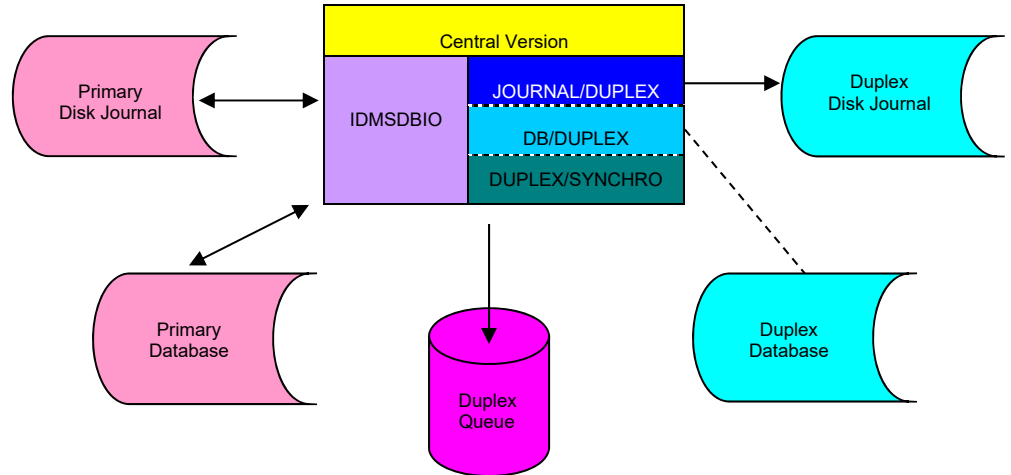


Figure 3 – Duplex Queuing

After the backup is completed, the duplex database will need to be recovered to the state of the primary. The duplex synchronization state is entered for this purpose. Figure 4 illustrates that during the resynchronization, both the duplex database and the duplex queue are being processed.

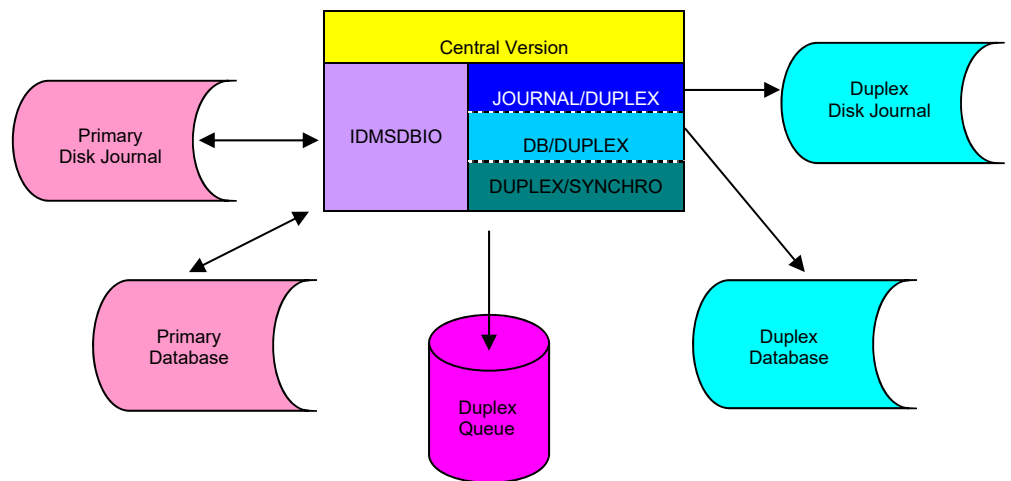


Figure 4 – Duplex Synchronization

SECTION 2: SYSTEM OPERATION

DUPLEX QUEUE

The online duplex synchronization process is an event driven type of process. The duplex database updates are dependent on the duplex queue update which only happens when there are update activities on the primary database. The updates are the events that drive the synchronization process.

When the synchronization process starts, it will update the duplex database at a faster rate than it does with the duplex queue. Given an update occurs during resynchronization, DUPLEX/SYNCHRO appends the new update in the back of the duplex queue, removes at least one previous update from the front of the queue, and applies the update to the duplex database. This process is repeated until the duplex queue is exhausted, at which time the synchronization will terminate and return to normal TDS operation.

The three stage processing cycle can be repeated indefinitely and thereby supporting a 24- hour CV operation with preventive maintenance. The setup preserves both IDMS and TDS operations, so duplex entities are not only used to prevent excessive downtime, but are also used as part of normal operation.

DUPLEX QUEUE

The 24X7 enhancement requires an extra external entity, the duplex queue. One duplex queue is needed for each CV that is to operate continuously. The duplex queue is used to preserve database integrity during the transition of TDS-CV processes from one to the other.

The duplex queue should be setup as two fixed length and unblocked QSAM files. The queue file blocksize should be 24 bytes longer than the largest blocksize of the CV. So, if a production CV's largest database file block is 9076, the duplex queue file's blocksize allocated for that CV should be 10000.

Sizing the duplex queue files is installation and time dependent; it is more difficult to accurately estimate the queue file size. Technetron recommends starting the duplex queuing process at low volume processing time in order to minimize the performance impact as well as keeping the queue size small. The queue files can grow large if the duplex queuing was started at a peak processing hour for an extended period. Since the queue will only grow during duplex queue processing, it would be a good time to check the size of the queue when duplex synchronization is first started.

The two queue files do not have to be the same size when they are allocated. It is recommended to allocate the first queue file to be slightly less than twice the size of the second queue file. So, if the allocation of the first queue file is 11 cylinders, it would be sufficient to assign 6 cylinders to the second queue file.

The duplex queue is attached to a CV by JCL specification; the DDname of the queue files determines the order of the queue files. Always assign the bigger allocation to the first queue file:

SECTION 2: SYSTEM OPERATION DUPLEX TABLE

```
//TDSQUEF1 DD DSN=FIRST.DUPLEX.QUEUE.FILE,DISP=SHR  
//TDSQUEF2 DD DSN=SECOND.DUPLEX.QUEUE.FILE,DISP=SHR
```

TDSQUEF1 is the DDname of the first duplex queue file, its size should be larger than that of TDSQUEF2. Since it is possible to allocate the queue files as new files at CV startup, a large secondary allocation is recommended if this practice is utilized.

The duplex queue is used to synchronize the duplex database with its primary counterpart. When a CV abnormally terminates during either duplex queuing or duplex synchronization, the duplex database is likely not to be the mirror image of the primary.

A batch duplex synchronization utility, TDSBSNC1, will read the duplex queue and apply its content to the duplex database. TDSBSNC1 will return a condition code of 4 when one of the two queue files is not emptied and a condition code of 8 when both queue files are not emptied.

Technetron recommends executing TDSBSNC1 before each start of the CV. The utility causes very little overhead and will terminate immediately if both queue files are emptied as in the case of normal CV shutdown. The execution JCL for TDSBSNC1 should be similar to that of the enhanced CV's JCL; it must have all the duplex database files and the two queue files.

DUPLEX TABLE

The online component can be easily customized to provide an installation with choices as to which database file, if any, and/or which disk journal file, if any, will be duplexed. The duplex table, DPLXTABL, is the entity which contains the duplex operation specifications defined and generated by the user.

DPLXTABL refers to a single occurrence of the duplex table, containing control information for a particular instance of operating environment. It is common for installations to create more than one DPLXTABL for the various environments' CVs. The file names (DDname) of the production IDMS systems often are not the same as the ones used in the test systems, therefore a DBA is likely to create two DPLXTABLs, one for each environment.

DPLXTABL is assembled at the installation and can be link-edited with the IDMSDBIO module or loaded as an independent module during duplexing's initialization. After it is loaded, the duplex table is searched to match each of the primary file name (DDname) and to derive the corresponding duplex DDname to ensure the duplex file will be properly processed.

The duplex table is also utilized to ensure database integrity for the 24 hour environment. Before the start of the duplex queuing process is started, the

SECTION 2: SYSTEM OPERATION

DUPLEX TABLE

queuing trigger matches all the duplex files with the corresponding DPLXTABL entries to check if the area(s) within a database file have been varied offline to preserve database integrity.

The first step of creating a DPLXTABL is to specify its content with the vendor supplied macros: **#DPLXHDR**, **#DPLXMAP**, and **#DPLXEND**.

#DPLXHDR is the duplex table header macro, it is used to specify whether the online component will process or bypass the database and the disk journal. The header contains duplex table identification and its release level. This macro is required in the duplex table definition.

Specify the **#DPLXHDR** macro as follows:

```
#DPLXHDR    [,DATA=BYPASS]  
             [,JOURNAL=BYPASS]
```

DATA=BYPASS is an optional parameter which specifies that all databases will be bypassed by DB/DUPLEX. This parameter means no duplex operation will be performed for any database and that no **#DPLXMAP** macro need to be specified for the databases.

JOURNAL=BYPASS is an optional parameter which specifies that all journals will be bypassed by JOURNAL/DUPLEX. This means no duplex operation will be performed for the disk journal and no **#DPLXMAP** macro will be necessary to specified for any of the disk journal files.

#DPLXMAP is the duplex table entry mapping macro; it is used for the specification of matching the duplex DDnames to the primary DDnames, and for the generation of the duplex file names. Each table entry will contain control information such as the method of update and the proper course of action after an I/O error for the associated file(s). The combination of the entries makes up the main body of a DPLXTABL.

Each **#DPLXMAP** requires two parameters, the first parameter represents the primary DDname(s) that will be examined by the online component's control module, the second parameter represents the duplex file name(s) the online component will use for the mirroring process. One or more of the macro must be specified. The order of specification is also important since the control module searches the duplex table sequentially.

SECTION 2: SYSTEM OPERATION DUPLEX TABLE

Specify the #DPLXMAP macro as follows:

#DPLXMAP DD=*primary*,DUPLXDD = { *duplex*
BYPASS }

[,MODE = { ACTIVE
PASSIVE ← }]

[,OPTIMIZE = { YES ←
NO }]

[,CHKAREA = { YES ←
NO }]

DD is a required parameter which specifies the name for one or more of the database or disk journal files that will be processed by the online component. *Primary* can be up to eight characters long and specifies a DDname in the DMCL. When the duplex software reads a DPLXTABL, it searches for a primary DDname which matches the DDname that IDMS is currently about to open or write to. During the search, the control module compares the IDMS DDname to the DPLXTABL *primary* DDname one character at a time, from the leftmost character to the rightmost. The *primary* DDname can use mask characters to specify a generic name which stands for multiple IDMS file names. The mask character is described as follows:

? = any alphabetic character

When the control module encounters any character in the IDMS DDname that does not match the corresponding character (position) in the *primary* DDname in the current entry of DPLXTABL, it reads the next entry in the duplex table and begins the comparison again. If none of the table entries provides a match, the duplex software will terminate with an user abend.

DUPLXDD is a required parameter which specifies the DDname of the file(s) that the online component will use to duplex. *Duplex* can be up to eight characters long and specifies a DDname that is not in the DMCL. Once the duplex control module finds a match in its search of the DPLXTABL, it will use the corresponding *duplex* as the duplex DDname or it will use the *duplex* in conjunction with the *primary* to derive the duplex DDname. The *duplex* DDname can contain mask characters with the mask character defined as follows:

SECTION 2: SYSTEM OPERATION

DUPLEX TABLE

? = any alphabetic character

When a *duplex* DDname has one or more mask characters, the duplex control module will substitute the corresponding character from the IDMS DDname (which is the DDname of the dataset that IDMS is currently processing with).

If DUPLXDD=BYPASS is coded, the online component's control module will bypass processing for the corresponding *primary* DD's. If #DPLXHDR has been coded so that all databases and/or the disk journal are to be processed by TDS (no DATA/JOURNAL=BYPASS is coded), then #DPLXMAP must be coded for every database and disk journal file. If duplex operation is not needed for a particular file, then code DUPLXDD=BYPASS on the entry mapping macro for that file.

Code a catchall DD parameter using all mask characters and the DUPLXDD=BYPASS as the last duplex table entry to avoid the user abend problem mentioned earlier. More importantly, since the DD parameters are searched sequentially and processed on a first match basis, follow the coding order of the most specific entries first and the most generic ones last. This practice will ensure TDS selects and process the right file.

MODE is an optional parameter which specifies how the online component will treat a duplex file if it is missing from the execution JCL during initialization. This parameter also governs the action to be taken after a file has experienced an I/O error.

If MODE=ACTIVE is coded, the duplex control module will treat the associated duplex file(s) as an active part of the duplexing process, and if one of the associated duplex files is missing from the startup JCL, then the online component terminates abnormally. Also, if an I/O error is detected on a duplex file marked as an active duplex file, the same result will occur.

When MODE=PASSIVE is coded for a duplex file that is missing from the execution JCL, the duplex control module will ignore the missing duplex file and it will act as if the corresponding primary file is not being duplexed. Also, if an I/O error is detected on a duplex file, then the online component will continue processing after taking a SNAP dump. The default of the MODE parameter is PASSIVE.

MODE=PASSIVE should be coded for the initial definition of the duplex table entry to enable the users to verify their execution JCL initially. The MODE=ACTIVE should only be coded for the critical files which users would not continue their operations without them being duplexed.

SECTION 2: SYSTEM OPERATION DUPLEX TABLE

OPTIMIZE is an optional parameter which specifies how the duplex file(s) will be processed. If OPTIMIZE=YES is coded, separate duplex buffers will be set up by the optimizer in order to realized the gain in performance throughput of the parallel I/O technique. When OPTIMIZE=NO is coded, IDMS buffers are used to perform the I/O operation. The default is OPTIMIZE=YES.

CHKAREA is an optional parameter which specifies whether or not to check all the areas within a file before starting duplex queuing. The default is CHKAREA=YES. Since some of the data dictionary areas and the areas of the system database work file cannot be varied offline once the CV is active, users should specify CHKAREA=NO for those files if they are being duplexed.

#DPLXEND is the duplex table footer macro; it marks the end of a duplex table with the time and date of its creation.

Specify the **#DPLXEND** macro as follows:

#DPLXEND

The coding of the three macros can be best demonstrated by an example. The following shows a sample DPLXTABL being defined using the duplex table definition macros.

EXAMPLE:

```
#DPLXHDR DD=D???DB,DUPLXDD=X???DB,CHKAREA=NO
#DPLXMAP DD=DB3???A,DUPLXDD=?????B,MODE=ACTIVE
#DPLXMAP DD=J?JRNL,DUPLXDD=?????XX,OPTIMIZE=YES
#DPLXMAP DD=????????,DUPLXDD=???????X
#DPLXEND
```

In this example, the DDnames would be transformed as follows:

<u>IDMS DDname</u>	<u>Duplex DDname</u>
DB3A1CA	DB3A1CB
DB4ATB2	DB4ATB2X
J1JRNL	J1JRNLXX
ABC	*ABC X
DMSGDB	XMSGDB

SECTION 2: SYSTEM OPERATION

TDS BATCH COMPONENT

- * This is an invalid DDname. The duplex table cannot be assembled because it cannot translate all the database and the disk journal files correctly, (#DPLXHDR macro was not coded with BYPASS for either the DATA or JOURNAL parameter). The DPLXTABL definition must be modified to allow for all DDnames in the DMCL. A possible modification would be to include an additional #DPLXMAP macro as follows:

```
#DPLXMAP DD=???,DUPLXDD=???X
```

and added as the first mapping entry in the above example.

After devising the desired definition of a duplex table, the next step is to make that DPLXTABL an operational entity. This is accomplished by using the system's linkage editor to install the assembled DPLXTABL object into IDMSDBIO or into a load module library. Sample JCLs for this procedure is supplied on the install tape. TDS's online component is functional with the successful completion of the above process.

TDS BATCH COMPONENT

TDS's batch component is made up of three utilities, ARCHIVE/DUPLEX, DUPLEX/SYNCHRO, and JOURNAL/DUPLEX's sequential journal mirroring facility. The batch component mirrors CA-IDMS's journal archive and its sequential (tape) journal. It also provides an efficient way to synchronize the online component's duplex entities with their counterparts.

By large, journal archives and sequential journals have been ignored. In the non-TDS IDMS environment, the typical DBA approach to recovery has been to backup the databases once a night, and let the journal archives take care of themselves. However, as the size of the databases grew and the nightly window shrunk, the archives have become more important. They provide the only way to recreate a database in recovery situations that calls for a rollforward or a rollback where one or more of the disk journal files has filled and been offloaded.

For very large, high activity databases that back up over the weekend, the journal archives become as important as CV's disk journal. Archives are the last resources to be used in recovery situations with minimum lost of data. The same is true with sequential journals, used mainly in large databases local mode operations for performance reason, their importance is not apparent until they are needed for recovery. At which time, it is the resource to recovery or to minimize the lost of data on the database.

The journal archive and the sequential journal should be offered the same protection as the database and the disk journal are offered with the online component. TDS's batch component ensures the availability of CA-IDMS archives

SECTION 2: SYSTEM OPERATION TDS BATCH COMPONENT

and sequential journals with its operating entities producing extra set(s) of those entities.

The batch component is consist of only static operating entities. Once install, it operates normally without further adjustments. The control for the batch component processing is done ONLY through JCL specifications.

The batch component attempts to open nine duplicate files with DDnames DUPJNL1 through DUPJNL9. If any of these files are present, it will write a copy of the journal archive or the sequential journal to each of the files. If no files are present then no duplicate entity will be created.

ARCHIVE/DUPLEX creates from one to nine duplicate journal archives every time the archive journal utility, IDMSAJNL, is executed to offload the IDMS disk journal. The selection of how many duplicates, if any, is made through JCL. The presence or absence of the DUPJNLn DDnames governs the duplication process.

Note that once the batch component is activated, it attempts to open the files with the predefined DDnames. The entire DD card should be omitted (not dummied out) if the corresponding duplicate file is not to be created.

Each of the desired duplicate archive DD cards must be added to the execution JCL for IDMSAJNL. Sample JCL for a DD card follows:

```

c                                                                 7
c 1..... 2
//dupjnl DD DSN=archive.dupjnl,
//      DISP=(,CATLG),UNIT=tape,
//      DCB=(RECFM=FB,LRECL=length,BLKSIZE=length)
```

Explanation:

dupjnl = One of DUPJNL1 through DUPJNL9
tape = Unit parameter for TAPE drives
length = Journal archive blocksize as specified in the DMCL

The sequential journal duplexing of JOURNAL/DUPLEX is controlled by the same process as described above. The same convention of DDnames and statements are used for the tape journal duplexing facility. The JCL used for the jobs that utilize this facility will be similar to the ones in the IDMSAJNL's JCL. The difference is likely to be that of the dataset name and the parameters of the data control block.

Make the JCL modifications to local mode jobs (the likely candidates to use the tape journal mirroring facility), and check the local DMCL setup to be executed with those jobs. The interest in the DMCL is because it contains the journal information which determines the disk journal or the sequential journal to be used.

SECTION 2: SYSTEM OPERATION

FILE PREPARATION AND USAGE

The tape journal mirroring facility is only active for the sequential journaling.

The batch component stand-alone utility, DUPLEX/SYNCHRO, copies IDMS database or disk journal files to their counterparts with great efficiency. Since the duplex entities are exactly the same as the primary, the utility can be used to copy the duplex files back to the primary's. Only JCLs are needed to operate DUPLEX/SYNCHRO. The entire definition of the utility JCLs is listed below:

```

c
c 1.....7
//stepname EXEC PGM=TDSUTIL1
//STEPLIB DD DSN=utility.loadlib,DISP=SHR
//PRMFILE1 DD DSN=primary.database.file1,DISP=OLD
//DPXFILE1 DD DSN=duplex.database.file1,DISP=OLD
//PRMFILE2 DD DSN=primary.database.file2,DISP=OLD
//DPXFILE2 DD DSN=duplex.database.file2,DISP=OLD
//PRMFILE3 DD DSN=primary.diskjrnl.file1,DISP=OLD
//DPXFILE3 DD DSN=duplex.diskjrnl.file1,DISP=OLD
//PRMFILE4 DD DSN=primary.diskjrnl.file2,DISP=OLD
//DPXFILE4 DD DSN=duplex.diskjrnl.file2,DISP=OLD
```

The template JCL shows the utility's load module, TDSUTIL1, will process four sets of database or disk journal files. It reads from the PRMFILEn DDs and writes to DPXFILEn DDs automatically. There are no parameters or control cards to setup; TDSUTIL1 will only process the set of files with both DD statements present and their dataset names specified.

TDSUTIL1 allocates and initializes its buffers according to the primary files blocksize. It issues multiple I/Os against all the files that are active. Maximum performance is realized by executing the utility to synchronize four set of files with equal or close number of blocks.

FILE PREPARATION AND USAGE

The batch component creates new duplex files every time it is executed. The new files are sequential datasets involving little or no work prior to their creation. TDS maintains the online duplex files once they are created the same way CA-IDMS does to its database and disk journal. For TDS's online component, allocation of the duplicate datasets must be done in exactly the same manner as allocating the primary datasets. That is, the file size, block length and dataset organization must equal those values specified in the primary dataset allocation process.

An online duplex dataset (database or disk journal) should be placed on a different disk volume from its corresponding primary dataset. This is to ensure that TDS operates more efficiently than otherwise possible and to not invalidate the duplex dataset in the event of the primary dataset disk becomes inoperative. TDS users also have the option to place the duplex database files on relatively higher load path than the primary files access path. Since the online component only writes

SECTION 2: SYSTEM OPERATION SYSTEM INTEGRATION

to its duplex datasets, the I/O counts against the duplex database are lower than those of the primary database. The optimal performance with TDS depends highly on the placement of the duplex files.

Initialization of the duplicate datasets can be accomplished in two ways. The first technique is to run the IDMS file initialization utility, IDMSINIT, against the duplicate datasets. This can be achieved by using the same IDMSINIT JCLs as was used in initializing the primary databases and disk journals, but changing the dataset name parameters to point to the duplicate datasets. Do not change the DDname when using IDMSINIT.

The second method is to ensure that normal operation is at a quiesce state (CVs have been shutdown normally, no database update activities), and then copy the primary databases and journals to the previously allocated duplex datasets. This second technique is the preferred approach since the duplicate datasets will be the exact copies of the primary entities. It is also the reasonable approach for installations that already are operating with IDMS as opposed to installations that are installing TDS and IDMS at the same time.

The duplex files can be viewed as immediate backups of their primaries; they are used primarily in recovery operations. The discussion of that usage is covered in the next section. The following covers the retrieval usage of the duplex entities.

Besides acting as backup files, the duplex files can be used to reduce contention on high volume systems. In both CV and local mode, the duplex files are not targets of any read operations, they have a lower I/O load comparing to their counterparts, and the duplex database can be used by other CVs or jobs for retrieval processing. For retrieval utilities like IDMSDBAN and IDMS journal reporter, substitute duplex databases and journals as the primary counterparts for reporting purpose.

The separation of operation also improves the security of the online system. Traditionally, the database files are shared through usage of the JCL disposition parameter. This setup allows access to a database from other sources, leaving the preservation of database integrity with IDMS only. With two sets of the same entity, CV can now enhance its access security by locking the primary files and sharing the duplex files.

SYSTEM INTEGRATION

The modularized components and the table-driven control mechanism of TDS provide users flexibilities in normal operations. TDS is likely to be implemented as a part of the online (CV) mode of operation, and some of the local mode and utility operations. The implementation process is accomplished by modifying the affected entity's JCLs, adding the duplex load library (concatenated ahead of IDMS's) and the duplex datasets.

SECTION 2: SYSTEM OPERATION

SYSTEM INTEGRATION

Once the integration is complete, CVs can be warmstarted. Local mode jobs and IDMS utilities can be run while TDS maintains the duplicate datasets as mirror image of their corresponding primary datasets which have, in turn, been specified in DPLXTABL. The duplex table and other TDS features such as the completeness of duplexing make the subsystem easy to maintain as well as it is to integrate with CA-IDMS. Additional integration may be needed for users who wish to take advantage of the presence of the duplex entities.

There are instances in local mode and in utility operation when TDS users may want to delay or bypass the updates to the duplex files until the operation is complete and the results are verified. Using a local mode job as an example, there are three ways to maintain a duplex database:

- Run the job without TDS against the primary dataset and repeat the effort for the duplex dataset.
- Run the job without TDS against the primary dataset and then copy the primary dataset over the duplex dataset.
- Run the job with TDS against the primary and the duplex datasets at the same time.

The first two options may take longer to run with at least double the amount of I/Os when compare to the third option which requires the same number of physical reads and two times the minimum number of physical writes, but they offered users the ability to process with two sets of equal entities, which in effect, may reduce the time of the overall operation, especially if sample job is only one part of a long batch process.

However, in the online production environment, some TDS users may want to restrict the options and operate continuously with full duplexing. They can accomplish this by defining the online duplex entities as active (MODE parameter of the #DPLXMAP macro) in the DPLXTABL and hard-link the duplex table with the rest of the TDS online component. This restrictive mode of operation ensures the synchronization of the primary entities with their duplex counterparts.

In other operations, such as normal development environment, the user may want to operate TDS with less rigidity. TDS can be flexible depending on how it was generated and how it is used. This flexibility, which may compromise the equality between the primary and the duplex entities, can be achieved by utilizing the dynamic duplex table load, passive mode of operation and dynamic file allocation/deallocation (DB/DYNAM, a separate product offered by Technetron).

An example of the least restrict mode of operation will be to generate the duplex table as an independent entity with all the online duplex files as passive, and warmstart the CV with the online component active but without any of the duplex datasets, operate for a period of time, quiesce and synchronize one part of the database and bring that part of the database back online with the protection of

SECTION 2: SYSTEM OPERATION SYSTEM INTEGRATION

TDS.

There are many setups between the most restrictive and the most flexible method of IDMS-TDS operations. TDS was designed to complement CA-IDMS's availability without compromising on performance or functionality; it has done so by providing users with great flexibilities and ease.

The subject of TDS integration with IDMS utilities can be compare to the local mode operation. As utilities supplied by the vendor, most CA-IDMS utilities dynamically load IDMSDBIO, and since TDS operates as an exit of IDMSDBIO, it is automatically activated.

However, there is a small number of utilities that are "hard linked" with IDMSDBIO, and even smaller number that do not use IDMSDBIO to access databases. These utilities include:

"hard-linked" : IDMSAJNL, IDMSDBN1, IDMSDUMP,
IDMSINIT, IDMSPFIX, IDMSRBCK,
IDMSRFWD, IDMSRSTR, RHDCTJIN

"no IDMSDBIO" : IDMSDIRL

An installation has three similar choices to the local mode jobs regarding the execution of these utilities:

- At execution time, run the utility against the primary (real) dataset and also run it against the duplex dataset,
- At execution time, run the utility against the primary (real) dataset and then copy the primary over the duplex dataset,
- At install time, relink the IDMS utilities that are "hard-linked" with IDMSDBIO so that they have duplexing capability. Then, at execution time, an IDMS utility need be run only once, and no copy from the primary to the duplex dataset is required.

The choice of technique depends on an installation's standards and the anticipated elapsed execution times of each approach. The first two choices require changes to utility execution JCL. The third choice requires no JCL change for utilities but the relinking of utility load modules may be required each time the duplex table is changed.

SECTION 2: SYSTEM OPERATION

SYSTEM INTEGRATION

Sample JCL for the third choice, relinking the utility load modules, follows:

```
c
c 1..... 7
//LKED EXEC PGM=IEWL,
// PARM='LET,LIST,MAP,XREF,
// SIZE=(196608,65536),NCAL'
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD UNIT=scratch-disk,SPACE=(CYL,6)
//SYSLMOD DD DSN=DUPLEX.TEST.LOADLIB,DISP=SHR
//LIB DD DSN=IDMS.TEST.LOADLIB,DISP=SHR
//SYSLIN DD *
REPLACE IDMSDBIO
INCLUDE LIB(utility)
INCLUDE SYSLMOD(IDMSDBIO)
ENTRY entry-name
NAME utility
```

where:

scratch-disk = unit parameter for scratch disk
utility = name of the utility load module being relinked
entry-name = entry point name for the utility being relinked

It is recommended that the third approach involving relinking the affected IDMS utilities is not chosen since the installer must ensure that all affected load modules are processed, and also since some utilities (e.g. IDMSDIRL) do not use IDMSDBIO and some utilities (e.g. IDMSINIT) do not always use IDMSDBIO although it is "hard linked" with the utilities themselves. The only utility which the third method should be implemented with is the journal archive utility, IDMSAJNL. Other utilities should be executed using one of the first two approaches.

IDMSAJNL should be relinked if TDS is used to mirror CV's disk journal. The relinked IDMSAJNL will ensure that both the primary and duplex journals are condensed together. Keeping CV's journals synchronized is a good operating practice and essential in recovery situations.

Last, TDS replaces the three IDMS database user exits which it utilize to enhance integration with other software. The IDMSAJNX, IDMSDPLX and IDMSJNL2 have been functionally replaced by TUDSAJNX, TUDSDPLX and TUDSJNL2 respectively. These TDS user exits require the same coding standards as the ones they replace, please refer to the IDMS documentation as to their restrictions.

DB/DYNAM

This section of the manual describes how to operate DB/DYNAM with CA-IDMS. First, the product functions are listed, then details of the functions and their parameters are discussed.

SECTION 2: SYSTEM OPERATION

DB/DYNAM

DB/DYNAM provides the means to allocate to and deallocate from a CA-IDMS Central Version's database, duplex database and VSAM files. It also performs certain duplex configurations dynamically, and serves as the control mechanism for duplex queuing and synchronization. It operates as an IDMS/DC task, and can perform the following:

- Allocate to CV an existing:
 - IDMS database that uses BDAM or VSAM as its access method,
 - IDMS native VSAM file,
 - non-IDMS physically sequential, direct access or native VSAM file,
 - duplex database for use by DB/DUPLEX,
- Close an opened:
 - IDMS database that uses BDAM or VSAM as its access method,
 - IDMS native mode VSAM file,
 - duplex database,
- Deallocate from CV a closed:
 - IDMS database that uses BDAM or VSAM as its access method,
 - IDMS native VSAM file,
 - non-IDMS physically sequential, direct access or native VSAM file,
 - duplex database created by DB/DUPLEX,
- Swap an IDMS database file with its duplex counterpart,
- Verify that all areas mapped to an opened database or IDMS file are offline prior to closing the database or file,
- Resume or suspend duplexing for a database file,
- Start duplex queuing or synchronization processing.

All file allocations, closings and deallocations remain in effect (unless superseded by subsequently issued DB/DYNAM commands) only during the current execution of the IDMS-CV/DC system; they have no effect on the definition of the DMCL or on the CV startup JCL. For each successful dynamic operation, DB/DYNAM outputs the associated message to the MVS and IDMS logs. These messages can be used to further integrate DB/DYNAM with IDMS through medication of the WTO exit.

DB/DYNAM utilizes IDMS-CV/DC line mode to communicate with a terminal and requires no other modules or user exit changes. It operates in all MVS environments that support SVC 99, and is compatible with CA-IDMS release 10.0 and above.

SECTION 2: SYSTEM OPERATION

DB/DYNAM

DB/DYNAM functions are requested by means of the DYNAM task. It is a set of online programs defined using normal sysgen parameters or it can be defined to CA-IDMS by using the DCMT VARY dynamic commands. Please refer to the installation section for the sysgen PROGRAM and TASK statements required to define the DYNAM task.

To request a DB/DYNAM function at run time, the user enters (in response to the ENTER NEXT TASK CODE prompt) the task code DYNAM followed by the appropriate command. If the task code and its parameters exceed 80 characters in length, enter the task code and DB/DYNAM will respond by redisplaying your input request allowing you to continue to enter more than 80 characters for your request. You can overtype, modify or expand your redisplayed input request as many times as you wish. At least one blank must separate the task code DYNAM and the command specification. Any number of blanks can separate the command and its parameters.

The DYNAM task can also be executed from a program. A sample program that illustrates dynamic operations performed at a set time is supplied with the installation tape. Many of the DYNAM command parameters correspond to JCL parameters. Further information on these parameters can be obtained from the IBM JCL Reference manual.

The remainder of this section presents the syntax and syntax rules, as well as a functional description and examples, for each of the DYNAM commands.

ALLOCATE COMMAND

The DYNAM ALLOCATE command dynamically locates and allocates an existing, cataloged dataset to IDMS-CV/DC. After dynamic allocation, areas within a CA-IDMS database or a CA-IDMS file dataset must be varied to either retrieval or update using the DCMT VARY AREA command. Syntax for the ALLOCATE command follows:

DYNAM ALLOCATE DDNAME *ddname-a* DSNAME *dsname-a*

**[MEMBER *member-a*] [DSORG { PS ←
VS
DA }]**

**[DISP { SHR ←
OLD }] [PASSWORD *password-a*]**

Syntax rules follow:

- **DDNAME *ddname-a*** specifies the ddname that is to be associated with the dynamically allocated dataset. For an IDMS database or IDMS file, *ddname-a* is the ddname specified in the file statement of the schema. For a DB/DUPLEX database, *ddname-a* is the ddname specified in the DPLXTABL table. For a non-IDMS file, *ddname-a* can be any valid ddname.
- **DSNAME *dsname-a*** specifies the name of the cataloged dataset associated with the ddname specified in the DDNAME parameter. The ddname is dummied out if the literal NULLFILE is used as the corresponding *dsname-a*.
- **MEMBER *member-a*** is used when the dataset named in the DSNAME parameter is a partitioned dataset. It specifies the particular member name that you wish to be associated with the ddname specified in the DDNAME parameter.
- **DSORG {PS/VS/DA}** specifies the dataset organization of the dataset specified in the DSNAME parameter. For a CA-IDMS database or a duplex database using BDAM as its access method, the dataset organization can be specified as either PS (physically sequential) or DA (direct access). For a CA-IDMS database using VSAM as its access method and for a native VSAM file, the dataset organization should be specified as VS (VSAM). The default is PS.
- **DISP {SHR/OLD}** specifies the disposition of the dataset specified in the DSNAME parameter. The default is SHR.
- **PASSWORD *password-a*** specifies the password associated with the dataset specified in the DSNAME parameter. If you specify the password, the system need not prompt the operator.

Examples of the DYNAM ALLOCATE command are given below:

Example 1: A cataloged BDAM database is dynamically allocated:

```
DYNAM A DD DBASE3A DSN EXAMPLE.DBASE3A
```

SECTION 2: SYSTEM OPERATION

DB/DYNAM - ALLOCATE

Example 2: A cataloged VSAM database is dynamically allocated with DISP=OLD and specifying its password:

```
DYNAM ALL DSN EXAMPLE.VSAM.DBASE4B DD
      DBASE4B DSORG VS DISP OLD PASS PSWD4B
```

NOTES

1. All necessary catalogs must be accessible to IDMS-CV/DC prior to using DB/DYNAM. The STEPCAT and JOBCAT dd's cannot be allocated or deallocated by DB/DYNAM. If the necessary catalogs are not available, then files that are cataloged in them cannot be located by DB/DYNAM and hence cannot be dynamically allocated to IDMS-CV/DC.
2. No IDMS journals can be allocated or deallocated by DB/DYNAM.
3. The STEPLIB and JOBLIB dd's cannot be allocated or deallocated by DB/DYNAM.
4. The CDMSLIB cannot be allocated or deallocated by DB/DYNAM.
5. The existence of the dataset specified in the DSNAME parameter is checked during the dynamic allocation process, but the existence of the member specified in the MEMBER parameter for a partitioned dataset is checked only when the file is opened.
6. A duplex database should be allocated either before the primary database is initially varied online or after the primary database is closed and before it is varied online again. This is because DB/DUPLEX begins creating a duplex database when CA-IDMS physically opens the primary database. If the duplex database is allocated after the primary database is physically opened, then DB/DUPLEX does not create a duplex database.
7. Direct access volumes containing datasets being allocated must be permanently mounted since a mount command is not issued by DB/DYNAM.

DEALLOCATE COMMAND

The DYNAM DEALLOCATE command dynamically deallocates a dataset from IDMS-CV/DC. The dataset must be closed prior to its dynamic deallocation. This can be accomplished by issuing an application program CLOSE request for non-IDMS/R files, by issuing a prior DYNAM CLOSE command, or by including the CLOSE parameter in the DYNAM DEALLOCATION command.

Syntax for the DEALLOCATE command follows:

DYNAM DEALLOCATE DDNAME *ddname-a* [CLOSE]

Syntax rules follow:

- **DDNAME *ddname-a*** specifies the ddname of the dataset that is to be dynamically deallocated.
- **CLOSE** specifies that DB/DYNAM is to close the dataset associated with the ddname specified in the DDNAME parameter prior to performing the dynamic deallocation. The CLOSE parameter applies to IDMS/R databases and IDMS/R files, and to DB/DUPLEX databases. The CLOSE parameter does not apply to non-IDMS/R files.

Examples of the DYNAM DEALLOCATE command are given below:

Example 1: A database is dynamically deallocated:

```
DYNAM DE DD DBASE3A
```

Example 2: A DB/DUPLEX database is dynamically deallocated:

```
DYNAM DE DD DUPLEX3A DSORG VS DISP OLD PASS  
PSWD4B
```

NOTES

1. The STEPCAT and JOBCAT dd's cannot be allocated or deallocated by DB/DYNAM.
2. IDMS/R journals cannot be allocated or deallocated by DB/DYNAM.
3. The STEPLIB and JOBLIB dd's cannot be allocated or deallocated by DB/DYNAM.
4. When DB/DUPLEX is used, both an IDMS/R database and its corresponding duplex database will be closed when CLOSED is specified and either the ddname of the IDMS/R database or the ddname of the duplex database is specified as ddname-a.
5. Prior to issuing this command with the CLOSE parameter, the areas contained in the database or IDMS/R file specified in the command must have been varied offline.

CLOSE COMMAND

The DYNAM CLOSE command closes IDMS/R databases, IDMS/R files and duplex databases created by DB/DUPLEX. This command is used prior to issuing the DYNAM DEALLOCATE command for IDMS/R databases, IDMS/R files and duplex databases. Non-IDMS/R files cannot be closed with DB/DYNAM.

Syntax for the CLOSE command follows:

DYNAM CLOSE DDNAME *ddname-a*

Syntax rules follow:

- **DDNAME *ddname-a*** specifies the ddname of the database or IDMS/R file that is to be closed. If *ddname-a* refers to an IDMS/R database for which DB/DUPLEX is creating a duplex copy or if *ddname-a* refers to a duplex database created as a copy of an IDMS/R database, then both the IDMS/R database and the duplex database are closed. This is to ensure that DB/DYNAM does not compromise the integrity of DB/DUPLEX.

This command performs the same function as the DCMT VARY FILE command of IDMS/R Release 10.2. The DCMT VARY FILE command should be used if the user is running in MPMODE.

Examples of the DYNAM CLOSE command are given below:

Example 1: An IDMS/R database is closed:

```
DYNAM C DD DBASE3A
```

Example 2: A native mode VSAM file (defined in the schema) is closed:

```
DYNAM C DD DBASEV1
```

Example 3: An IDMS/R database (file name DBASEZ1) that is being duplicated by DB/DUPLEX (using file name DBDUPZ1) is closed:

```
DYNAM CLOSE DDNAME DBASEZ1
```

Note that both file (DBASEZ1 and DBDUPZ1) are closed, and that either of the file names could have been used in the CLOSE command.

SECTION 2: SYSTEM OPERATION

DB/DYNAM - CLOSE

NOTES

1. IDMS/R journals cannot be closed by DB/DYNAM.
2. Non-IDMS/R files (i.e., files that are not defined in the schema DMCL or in the DPLXTABL table) cannot be closed by DB/DYNAM.
3. Prior to issuing a CLOSE, SWAP, or START for a particular database file, all areas contained in the database or IDMS/R file specified in the command must have been varied offline.
4. The STEPCAT/JOBCAT/STEPLIB/JOBLIB dd's cannot be closed by DB/DYNAM.

SWAP COMMAND

The DYNAM SWAP command exchanges the DCB anchor address of the primary file with that of the duplex file. When a primary database file has been swapped with its counterpart, the operation has the equivalent effect of physically switching the two files, I/Os issued against the primary file will be performed on the duplex file and vice versa. The corresponding area(s) within the file being swapped must be offline in order for the command to complete successfully.

Syntax for the SWAP command follows:

DYNAM SWAP DDNAME *ddname-a*

Syntax rules follow:

- **DDNAME *ddname-a*** specifies the ddname of the database or IDMS/R file that is to be swapped. If *ddname-a* refers to an IDMS/R database for which DB/DUPLEX is creating a duplex copy or if *ddname-a* refers to a duplex database created as a copy of an IDMS/R database, then both the IDMS/R database and the duplex database are swapped. This is to ensure that DB/DYNAM does not compromise the integrity of DB/DUPLEX.

This command performs the same function as the DCMT VARY FILE command of IDMS/R Release 10.2. The DCMT VARY FILE command should be used if the user is running in MPMODE.

Example of the DYNAM SWAP command is given below:

Example 1: An IDMS/R database is swapped:

```
DYNAM SW DD DBASE3A
```

SECTION 2: SYSTEM OPERATION

DB/DYNAM - DUPLEX

DUPLEX COMMAND

The DYNAM DUPLEX command starts or resume duplex operation for a database which has been previously bypassed either in the DPLXTABL generation or when using the DYNAM BYPASS command. The corresponding area(s) within the file being duplex must be offline in order for the command to complete successfully.

Syntax for the DUPLEX command follows:

DYNAM DUPLEX DDNAME *ddname-a*

Syntax rules follow:

- **DDNAME *ddname-a*** specifies the ddname of the database or IDMS/R file that is to be duplexed. If *ddname-a* refers to an IDMS/R database for which DB/DUPLEX is creating a duplex copy or if *ddname-a* refers to a duplex database created as a copy of an IDMS/R database, then both the IDMS/R database and the duplex database are duplexed. This is to ensure that DB/DYNAM does not compromise the integrity of DB/DUPLEX.

Example of the DYNAM DUPLEX command is given below:

Example 1: An IDMS/R database file duplex operation is started:

```
DYNAM DUP DD DBASE3A
```

BYPASS COMMAND

The DYNAM BYPASS command suspend the duplex operation for the file specified. The corresponding area(s) within the file being bypassed must be offline in order for the command to complete successfully.

Syntax for the BYPASS command follows:

DYNAM BYPASS DDNAME *ddname-a*

Syntax rules follow:

- **DDNAME *ddname-a*** specifies the ddname of the database or IDMS/R file that is to be bypassed. If *ddname-a* refers to an IDMS/R database

for which DB/DUPLEX is creating a duplex copy or if *ddname-a* refers to a duplex database created as a copy of an IDMS/R database, then both the IDMS/R database and the duplex database are bypassed. This is to ensure that DB/DYNAM does not compromise the integrity of DB/DUPLEX.

Example of the DYNAM BYPASS command is given below:

Example 1: An IDMS/R database file duplex operation is bypassed:

```
DYNAM BY DD DBASE3A
```

START COMMAND

The DYNAM START command is used to start the duplex queuing or the duplex synchronization process. All areas defined in the DPLXTABL with the CHKAREA=YES parameter are required to be offline before DB/DYNAM performs the duplex queuing initialization.

Syntax for the START command follows:

```
DYNAM START { QUEUE  
                  SYNCHRO }
```

Syntax rules follow:

- There must be a processing option of either QUEUE or SYNCHRO. There is no default or other parameters for the START command.

Examples of the DYNAM START command are given below:

Example 1: Starting duplex queuing:

```
DYNAM START Q
```

Example 2: Starting duplex synchronization:

```
DYNAM ST SYN
```

DB/INFO COMMAND

DB/INFO is a set of IDMS-DC modules that display information related to IDMS and duplex operations. The modules utilizes IDMS-DC line mode to output the query information.

SECTION 2: SYSTEM OPERATION

DB/INFO

There are six parameters that DB/INFO processes, three for IDMS-DC, three for duplex processing. The duplex parameters are prefixed by DPLX. The parameters are passed to DB/INFO on a command line.

Syntax for the DB/INFO command follows:

- INFO DATABASE** Lists the database files, the corresponding dataset and their status.
- INFO DPLXFILE** Lists the duplex files, database and journal, the corresponding dataset and duplexing status.
- INFO DPLXQUE** Returns the mode of XPO status, normal, queuing or synchronizing.
- INFO DPLXTABL** Shows the duplex table online.
- INFO JOURNAL** Lists the journal files, the corresponding dataset and their status.
- INFO LOADLIB** Lists the steplib and the CDMSLIB with their corresponding datasets attached to this system.

The following are sample output screens for the DB/INFO command. For DB/INFO database parameter:

DB/INFO		Technetron Dynamic Facility		Release 3.0	
		© Copyright Technetron Unlimited, Inc. 1993.			
DATA					
DICTDB	IDMS.TEST.DICTDB	SHR	OPEN	PS	
DMSGDB	IDMS.TEST.DMSGDB	SHR	OPEN	PS	
DLODDB	IDMS.TEST.DLODDB	SHR	OPEN	PS	
DLOGDB	IDMS.TEST.DLOGDB	SHR	OPEN	PS	
DRUNDB	IDMS.TEST.DRUNDB	SHR	OPEN	PS	
DSCRDB	IDMS.TEST.DSCRDB	SHR	OPEN	PS	
DQUEDB	IDMS.TEST.DQUEDB	SHR	OPEN	PS	
EMPDEMO	IDMS.TEST.EMPDEMO	SHR	OPEN	PS	
INSDEMO	IDMS.TEST.INSDEMO	SHR	OPEN	PS	
ORGDEMO	IDMS.TEST.ORGDEMO	SHR	OPEN	PS	
USER01DB	IDMS.TEST.USER01DB	SHR	OPEN	PS	
USER02DB	IDMS.TEST.USER02DB	SHR	OPEN	PS	
USER03DB	IDMS.TEST.USER03DB	SHR	OPEN	PS	
USER04DB	IDMS.TEST.USER04DB	SHR	OPEN	PS	
USER05DB	IDMS.TEST.USER05DB	SHR	OPEN	PS	
PAGE 001 – NEXT PAGE:					

**SECTION 2: SYSTEM OPERATION
DB/INFO**

For DB/INFO duplex file parameter:

```

DB/INFO                               Technetron Dynamic Facility                Release 3.0
                                       © Copyright Technetron Unlimited, Inc. 1993.

                                     DPLXF

    DICTDB      DUPLEX -----
    XICTDB      IDMSRTS.DBA.REGION.XICTDB                                SHR   OPEN   PS

    DMSGDB      DUPLEX ----- I/O ERROR -----
    XMSGDB      IDMSRTS.DBA.REGION.XMSGDB                                SHR                                PS

    DLODDB      DUPLEX -----
    XLODDB      *** DDNAME MISSING ***

    DLOGDB      DUPLEX -----
    XLOGDB      *** DDNAME MISSING ***

    DRUNDB      DUPLEX -----
    XRUNDB      *** DDNAME MISSING ***
                                     PAGE 001 - NEXT PAGE:

```

There are three sample output screen shown for the duplex queue parameter, one for each duplex processing state.

```

DB/INFO                               Technetron Dynamic Facility                Release 3.0
                                       © Copyright Technetron Unlimited, Inc. 1993.

                                     DPLXQ

                                     DUPLEX OPERATION - NORMAL

    V99 ENTER NEXT TASK CODE:

```

SECTION 2: SYSTEM OPERATION
DB/INFO

Sample output layout when duplex queuing is active:

```

DB/INFO                               Technetron Dynamic Facility           Release 3.0
                                     © Copyright Technetron Unlimited, Inc. 1993.

      INFO DPLXQ

      DUPLEX OPERATION – QUEUING  QUEUE FILE-ONE  RECORD COUNT: 0000159

V99 ENTER NEXT TASK CODE:
  
```

Sample screen layout when duplex synchronization is active:

```

DB/INFO                               Technetron Dynamic Facility           Release 3.0
                                     © Copyright Technetron Unlimited, Inc. 1993.

      DPLXQ

      DUPLEX OPERATION – (RE)SYNC  QUEUE FILE-ONE  RECORD COUNT: 0000563
                                     QUEUE FILE-TWO  RECORD COUNT: 0000238

V99 ENTER NEXT TASK CODE:
  
```

For the duplex table parameter:

```

DB/INFO                               Technetron Dynamic Facility           Release 3.0
                                     © Copyright Technetron Unlimited, Inc. 1993.

      DPLXT

DPLXTABL          BYPASS JOURNALS: NO          BYPASS DATABASES: NO

PRIMARY DDNAME    DUPLEX DDNAME      MODE    OPTIMIZE    CHK AREA
D???DB           X???DB        PASSIVE YES        NO
USER???DB        DPLX???DB    PASSIVE YES        YES
J?JRNL          X?JRNL       PASSIVE YES        YES
????????        BYPASS       PASSIVE YES        YES

V99 ENTER NEXT TASK CODE:
  
```

SECTION 2: SYSTEM OPERATION DB/INFO

For the loadlib parameter:

DB/INFO	Technetron Dynamic Facility © Copyright Technetron Unlimited, Inc. 1993.	Release 3.0
LOAD		
STEPLIB	IDMS.TEST.CV99.LOADLIB	SHR PO
CDMSLIB	IDMS.TEST.CV99.LOADLIB	SHR PO
	IDMS.TEST.DPLX.LOADLIB	SHR ??
	IDMS.TEST.APPL.LOADLIB	SHR ??
	IDMS.TEST.SYSTEM.LOADLIB	SHR ??
V99 ENTER NEXT TASK CODE:		

For the journal parameter:

DB/INFO	Technetron Dynamic Facility © Copyright Technetron Unlimited, Inc. 1993.	Release 3.0
JOURNAL		
J1JRNL	IDMS.TEST.J1JRNL	SHR OPEN PS
J2JRNL	IDMS.TEST.J2JRNL	SHR OPEN PS
J3JRNL	IDMS.TEST.J3JRNL	SHR OPEN PS
SYSJRNL	*** DDNAME MISSING ***	SHR OPEN PS
V99 ENTER NEXT TASK CODE:		

RECOVERY OPERATION

TDS's flexibility and the options the subsystem generates have resulted in various improvements to the IDMS's normal processing environment. But the main purpose of running TDS is to minimize the devastation caused by the unavailability of any of the CA-IDMS external entities. In ensuring the availability of the recovery resource needed by the IDMS recovery facility greatly enhances the DBMS's recovery operation, and remains as the TDS primary function.

We refer to the failure in accessing the IDMS external entities as an I/O error. The I/O error is considered to be "hard" when the target entity of the I/O operation is inaccessible to IDMS due to equipment (hardware) failure. Under this circumstance and with the protection of TDS, the recovery procedure can use the duplex entity to recover from the error so IDMS can resume normal operation quickly.

The TDS approach to I/O error recovery is described in this section. It is divided into CV, local mode and utility recovery operations. Each recovery operation includes generalized procedures which can be used as the guide to different installation's customized recovery procedures.

CENTRAL VERSION RECOVERY

In the event that an I/O error occurs on a CV's database or its journal, the installation should restart CV, if CV was running, to determine whether or not the I/O error is "hard". If the error is not "hard", CV will successfully warmstart and the recovery procedure that an installation should perform is the same as that used in a normal warmstart-after-crash situation.

The recent CA-IDMS enhancements have included some I/O error handling capability, a CV experiencing I/O difficulties on its database may keep on operating without abnormal termination. However, this does not resolve the problem of data loss on the error file(s), since the cause of error may relate to other hardware unit such as the disk controller or the damage to the database file may not be confined to one segment of the dataset.

The same approach was taken in handling CV's disk journal error. Since users usually generate the disk journal with multiple files, IDMS was enhanced to automatically switch to the next disk journal file when it has detected an I/O problem on the present journal file, the DBMS may also vary the affected database areas offline making them inaccessible. Here again, without determining the cause of the error and the scope of the problem, the user may be left in a worst state than if the online system was brought down immediately, recovered with the duplex entities, and restarted to resume normal operation quickly.

The TDS I/O error handler avoids the dilemma which the IDMS I/O error routine faces. It offers the user the option to terminate online processing abnormally or continue CV's normal operation and bypass the problem entity from the point of

SECTION 3: RECOVERY OPERATION

CV DATABASE RECOVERY PROCEDURE

error. The recovery procedure for the first option is the same as the ones described later on in this section, CV database recovery procedure and CV journal recovery procedure. The second option eliminates the need for the immediate invocation of the recovery operation. By delaying the recovery, the normal synchronization process (described under File Preparation and Usage in Section 2) of primary and duplex entities will be performed at a later time.

The CV recovery operation with TDS is separated into two procedures, one for database and one for disk journal. The technique described by the procedures is the recommended approach to CV recovery in the event of a "hard" I/O error occurring on the database or the disk journal file protected by TDS. For database files and disk journal files not protected by TDS, your current installation recovery procedure covering I/O errors should be followed.

CV Database Recovery Procedure

It is important to keep in mind that the proper identification of the entity in error must be made prior to the execution of the recovery procedure. The CV database recovery procedure is made of several steps, each of which is described here:

- Modify the startup JCL
 - Warmstart CV
 - Shutdown CV
 - Recreate the failed database
 - Restore the startup JCL
1. Modify the startup JCL - If the I/O error occurred on the primary file, then for that database file, alter its DD card in the startup JCL to use its duplicate dataset (maintained by TDS) rather than the primary dataset and also dummy out the duplicate file's DD. For example,

Before modification:

```
//DB4ATB DD DSN=IDMS.PROD.DB4ATB,DISP=OLD  
//DB4ATBX DD DSN=IDMS.PROD.DUPLEX.DB4ATB,DISP=SHR
```

After modification:

```
//DB4ATB DD DSN=IDMS.PROD.DUPLEX.DB4ATB,DISP=OLD  
//DB4ATBX DD DUMMY
```

SECTION 3: RECOVERY OPERATION

CV JOURNAL RECOVERY PROCEDURE

If the I/O error occurred on the duplicate file, then the DD card for that file should be dummied out. For example,

Before modification:

```
//DB4ATB DD DSN=IDMS.PROD.DB4ATB,DISP=OLD
//DB4ATBX DD DSN=IDMS.PROD.DUPLEX.DB4ATB,DISP=SHR
```

After modification:

```
//DB4ATB DD DSN=IDMS.PROD.DB4ATB,DISP=OLD
//DB4ATBX DD DUMMY
```

2. Warmstart CV - Perform a warmstart on CV. If it warmstarts successfully, then continue with this procedure. If it does not warmstart successfully, then discontinue this procedure, and perform your installation's current recovery procedure for an I/O error.
3. Shutdown CV - Depending on your installation's standards, the criticality of the data in the affected database and the amount of downtime that is acceptable, shutdown CV normally either at the end of the day's processing or as soon as the warmstart completes. Now offload the journals in the normal manner.
4. Recreate the failed database - The database file that suffered the I/O error must be reallocated and then recreated by copying from the duplex database file. Now, perform your normal database backup procedure.
5. Restore the startup JCL - Reverse the JCL modification done in Step 1 so that the DD for the primary database points to the primary database file, and the DD for the duplicate database points to the duplicate database file. If you shutdown CV in Step 3 as soon as CV warmstarted successfully, then you can now restart CV and continue normal processing.

This recovery procedure can be executed with the most speed by leaving CV up after the Step 2 warmstart until the normal shutdown time. This means that the downtime is minimized to the time needed to identify the I/O error and the file it affects, perform the JCL modification and do a CV warmstart.

CV Journal Recovery Procedure

It is important to keep in mind that the proper identification of the entity in error must be made prior to the execution of the recovery procedure. The CV journal recovery procedure is made of several steps, each of which is described here:

SECTION 3: RECOVERY OPERATION

CV JOURNAL RECOVERY PROCEDURE

- Modify the startup JCL
 - Warmstart CV
 - Shutdown CV
 - Recreate the failed journal
 - Restore the startup JCL
1. Modify the startup JCL - If the I/O error occurred on the primary journal, then for that journal, alter its DD card in the startup JCL to use its duplicate dataset (maintained by TDS) rather than the primary dataset and also dummy out the duplicate file's DD. For example,

Before modification:

```
//J1JRNL DD DSN=IDMS.PROD.J1JRNL,DISP=SHR
//J1JRNLXX DD DSN=IDMS.PROD.DUPLEX.J1JRNL,DISP=SHR
```

After modification:

```
//J1JRNL DD DSN=IDMS.PROD.DUPLEX.J1JRNL,DISP=SHR
//J1JRNLXX DD DUMMY
```

If the I/O error occurred on the duplicate journal, then the DD card for that file should be dummied out. For example,

Before modification:

```
//J1JRNL DD DSN=IDMS.PROD.J1JRNL,DISP=SHR
//J1JRNLXX DD DSN=IDMS.PROD.DUPLEX.J1JRNL,DISP=SHR
```

After modification:

```
//J1JRNL DD DSN=IDMS.PROD.J1JRNL,DISP=SHR
//J1JRNLXX DD DUMMY
```

2. Warmstart CV - Perform a warmstart on CV. If it warmstarts successfully, then continue with this procedure. If it does not warmstart successfully, then discontinue this procedure, and perform your installation's current recovery procedure for an I/O error.
3. Shutdown CV - Depending on your installation's standards, the criticality of the affected data and the amount of downtime that is acceptable, shutdown CV normally either at the end of the day's processing or as soon as the warmstart completes. Now offload the journals in the normal manner.

SECTION 3: RECOVERY OPERATION

LOCAL MODE RECOVERY

4. Recreate the failed journal - The journal that suffered the I/O error must be reallocated and then recreated by copying the duplex journal. Now, perform your normal database backup procedure.
5. Restore the startup JCL - Reverse the JCL modification done in Step 1 so that the DD for the primary disk journal points to the primary disk journal file, and the DD for the duplex disk journal points to the duplicate disk journal file. If you shutdown CV in Step 3 as soon as CV warmstarted successfully, then you can now restart CV and continue normal processing.

This recovery procedure can be executed with the most speed by leaving CV up after the Step 2 warmstart until the normal shutdown time. This means that the downtime is minimized to the time needed to identify the I/O error and the file it affects, perform the JCL modification and do a CV warmstart.

LOCAL MODE RECOVERY

There are more options and complexity involved with the local mode recovery operation than that of CVs. The main reason for this is the lack of system restart capability or the equivalent of the CV's automatic recovery facility. That automatic recovery facility preserves the physical data integrity by rolling back the unfinished database work units at the time of a failure; the logical data integrity is also preserved by the same process due to the nature of online processing. A successfully warmstart (after a failure) CV's database will accurately reflect all the completed database work up to the point of that failure. Since online applications segment their database work according to the functions they support, they are often kept concise and in logical units helping to preserve the logical database integrity in the recovery process.

The repetitive execution using the same source of input by local mode jobs represents the nature of its normal operation, batch processing. Local mode recovery operation depends on the manual recovery utilities to preserve the database physical integrity. However, the preservation of logical data integrity rest solely on the application and can be burdensome for some of the more complicated batch processing. That preservation for applications to resume their processing after a stoppage is referred to as the application restartability.

The recovery procedure of a local mode application depends largely on its restart capability and the amount of data processed up to the point of failure. Journaling is required during local mode operation in order for users to retain the restart capability. Since the manual recovery utilities can only function properly with portable storage media, it is common to journal directly to tapes or tape cartridges during normal operation. Due to the reasons stated above, IDMS users have been simplifying their local mode operation by foregoing the restart capability. Instead, they restore and reprocess their local mode jobs when unexpected interruptions occur.

SECTION 3: RECOVERY OPERATION

LOCAL MODE DATABASE RECOVERY PROCEDURE

IDMS-TDS users with full online database duplexing who have chosen to deactivate TDS during normal local mode processing without journaling may have the most to gain in their recovery operation. Their local mode recovery procedure would only be that of the databases, and it should be as simple as copying the set of files (primary or duplex) that was not used in the failed operation back onto the other set.

In the case of database I/O error recovery involving local mode update (to one set of database files) jobs without journaling, the optional step of reallocating the problem file should be performed before executing the synchronization process as described in the previous paragraph. If the files have been backed-up prior to the execution of the local mode job, the user can simply rerun the same job using the other set of database files while reallocating the problem file and restore the set of database files that was used in the failed operation.

Some local mode applications reserve their restart capability because of the volume of data which they process or the size of the database they are processed with. A long running local mode job that processed a high number of transactions before a failure occurred, or a local mode job updating a large database when encountering problems, should have their database recovered up to the point of failure and restarted if the recovery operation takes less time than the restore and reprocess procedure. With protection of TDS, the recoverability is insured and therefore the physical database integrity can be preserved when I/O errors are encountered.

For users who retain their local mode restart capability and for those who operate with TDS in local mode, they have more options in their execution of the recovery procedure. To simplify matters, we will limit our discussion of local mode recovery to operation with TDS for an I/O error. For entities not protected by TDS, the current installation recovery procedure covering I/O errors should be followed.

Local Mode Database Recovery Procedure

It is important to keep in mind that the proper identification of the entity in error must be made prior to the execution of the recovery procedure. The local mode database recovery procedure is made of several steps, each of which is described here:

- Modify the application JCL
- Perform manual recovery
- Rerun/restart the application
- Recreate the failed database
- Restore the application JCL

SECTION 3: RECOVERY OPERATION

LOCAL MODE DATABASE RECOVERY PROCEDURE

1. Modify the application JCL - This is an optional step, skip this step if TDS was installed into the manual recovery utilities (see Step 2 and Step 3). If the I/O error occurred on the primary file, then for that database file, alter its DD card in the JCL to use its duplicate dataset (maintained by TDS) rather than the primary dataset and also dummy out the duplicate file's DD. For example,

Before modification:

```
//DD4ATB DD DSN=IDMS.PROD.DB4ATB,DISP=OLD
//DB4ATBX DD DSN=IDMS.PROD.DUPLEX.DB4ATB,DISP=OLD
```

After modification:

```
//DB4ATB DD DSN=IDMS.PROD.DUPLEX.DB4ATB,DISP=OLD
//DB4ATBX DD DUMMY
```

If the I/O error occurred on the duplicate file, then the DD card for that file should be dummied out. For example,

Before modification:

```
//DB4ATB DD DSN=IDMS.PROD.DB4ATB,DISP=OLD
//DB4ATBX DD DSN=IDMS.PROD.DUPLEX.DB4ATB,DISP=OLD
```

After modification:

```
//DB4ATB DD DSN=IDMS.PROD.DB4ATB,DISP=OLD
//DB4ATBX DD DUMMY
```

2. Perform manual recovery - Perform the normal "journal offload/JFIX/ rollback-inflight-tasks" sequence as described in IDMS local mode recovery discussion. At this point, installations operating with TDS in local mode, have the option of executing Step 4 without the backup, and then execute this step to recover the two databases together. If this approach is taken, then the procedure is simplified, and performed in the order of Step 4 (no backup), Step 2, backup, and execute Step 3 as it was part of normal operation. If not, execute this step and continue.
3. Rerun/restart the application - Depending on the recovery performed up to this step (see Step 2), the installation standards, the criticality of the data in the affected database and the length of delay that is acceptable, this may be the last step of the recovery procedure. If TDS was not installed or not used to recover the erroneous database prior to this step, you still have the option of starting at Step 2, execute Step 4 as is, and then execute this step as part of normal operation using the unchanged JCL. Otherwise rerun/restart the application with the modified JCL here or go to Step 4.

SECTION 3: RECOVERY OPERATION

LOCAL MODE JOURNAL RECOVERY PROCEDURE

4. Recreate the failed database - The database file that suffered the I/O error must be reallocated and then recreated by copying the duplex database file, then perform the normal database backup procedure.
5. Restore the application JCL - Reverse the JCL modification done in Step 1 so that the DD for the primary database points to the primary database file, and the DD for the duplicate database points to the duplicate database file. If you have not done so, then you can now rerun/restart the application.

Local Mode Journal Recovery Procedure

It is important to keep in mind that the proper identification of the entity in error must be made prior to the execution of the recovery procedure. The local mode journal recovery procedure is made of several steps, each of which is described here:

- Modify the application JCL
 - Perform manual recovery
 - Rerun/restart the application
 - Recreate the failed journal
 - Restore the application JCL
1. Modify the application JCL - This is an optional step, skip this step if local mode journaling has been directed to tapes or cartridges. If the I/O error occurred on the primary journal, then for that journal, alter its DD card in the JCL to use its duplicate dataset (maintained by TDS) rather than the primary dataset and also dummy out the duplicate file's DD. For example,

Before modification:

```
//J1JRNL DD DSN=IDMS.PROD.J1JRNL,DISP=OLD  
//J1JRNLXX DD DSN=IDMS.PROD.DUPLEX.J1JRNL,DISP=OLD
```

After modification:

```
//J1JRNL DD DSN=IDMS.PROD.DUPLEX.J1JRNL,DISP=OLD  
//J1JRNLXX DD DUMMY
```

SECTION 3: RECOVERY OPERATION

UTILITY RECOVERY

If the I/O error occurred on the duplicate journal, then the DD card for that file should be dummied out. For example,

Before modification:

```
//J1JRNL DD DSN=IDMS.PROD.J1JRNL,DISP=OLD
//J1JRNLXX DD DSN=IDMS.PROD.DUPLEX.J1JRNL,DISP=OLD
```

After modification:

```
//J1JRNL DD DSN=IDMS.PROD.J1JRNL,DISP=OLD
//J1JRNLXX DD DUMMY
```

2. Perform manual recovery - Perform the normal "journal offload/JFIX/ rollback-inflight-tasks" sequence as described in IDMS local mode recovery discussion. Start with this step if tape journal was used. Depending on if TDS was installed with the recovery utilities, execute this step with the "good" tape journal to recover one or both sets of the database, synchronize the two databases (optional if TDS not active in local mode), and continue with Step 3 as part of normal operation. If not, execute this step and continue.
3. Rerun/restart the application - Depending on the recovery performed up to this step, the installation standards, the criticality of the data in the affected database and the length of delay that is acceptable, this may be the last step of the recovery procedure. If the local mode job tape journaled, this is the continuation of normal processing. Otherwise, rerun/restart the application with the modified JCL here or go to Step 4.
4. Recreate the failed journal - The journal that suffered the I/O error must be reallocated and then recreated by copying the duplex journal.
5. Restore the application JCL - Reverse the JCL modification done in step 1 so that the DD for the primary journal points to the primary journal, and the DD for the duplicate journal points to the duplicate journal. If you have not done so, then you can now rerun/restart the application.

UTILITY RECOVERY

The IDMS utility performance depends on the individual utility and its capacity. Normally, most IDMS utilities are operationally equivalent to local mode applications, their recovery operations then are also similar to that of the local mode jobs including the high dependency on the normal operating environment. The simple procedure of restore and reprocess is usually the only steps needed for retrieval utilities in the case of I/O failures or simply rerun the same utility against the other set of entity for TDS users. Here again, we need to limit ourselves to the update utilities operating

SECTION 3: RECOVERY OPERATION

IDMSAJNL JOURNAL RECOVERY PROCEDURE

in the IDMS-TDS environment, and focus the utility recovery on I/O error occurring on the entity which they maintain or create.

It is more convenient to discuss the utility recovery operation by making the distinction between restartable and non-restartable utilities due to their similarity to local mode applications. The non-restartable ones such as the database load utility, IDMSDBLU, operate in an all-or-nothing mode, their I/O recovery procedure would be to recreate the error file and rerun the utility. The recovery procedure of the restartable utilities like the logical delete utility (IDMSLDEL), would be similar or the same as the local mode recovery procedures described in the previous part of this section.

The exception of the similarities between local mode jobs and IDMS utilities is the archive journal utility, IDMSAJNL. It differs operationally in its creation and maintenance of external entities comparing to other local mode jobs. However, its recovery procedures can be easily derived from substituting the different entities of the local mode procedures.

The presumption of operating procedures and of recovery methodology eliminates the various options available for the recovery procedures. The assumptions include 1) both ARCHIVE/DUPLEX and JOURNAL/DUPLEX have been installed into IDMSAJNL, 2) the normal archive operation is require to produce two sets of journal archive and condense both the primary and the duplex disk journals simultaneously in one execution of the utility, and 3) no compromises of the second requirement during recovery operation. Depending on the occurrence of the I/O error and the entity which it affects, two procedures are needed as part of IDMSAJNL recovery operation, one procedure is for recovering from disk journal error, and the other for the recovery of journal archive I/O failures.

IDMSAJNL Journal Recovery Procedure

It is important to keep in mind that the proper identification of the entity in error must be made prior to the execution of the recovery procedure. The IDMSAJNL journal recovery procedure is made of several steps, each of which is described here:

- Modify the input parameters/JCL
- Recreate the failed journal file
- Restart the utility
- Restore the input parameters/JCL

SECTION 3: RECOVERY OPERATION

IDMSAJNL ARCHIVE RECOVERY PROCEDURE

1. Modify the input parameters/JCL - IDMSAJNL is a restartable utility. Since the input parameter is to control all processing, it must be changed to reflect the desired operation. For example,

Before modification:

```
DMCL=CV99DMCL,OFFLOAD=YES,BUFFERS=10
```

After modification:

```
DMCL=CV99DMCL,FILE=J2JRNL,OFFLOAD=RESTART
```

It is not uncommon to keep the different sets of IDMSAJNL input parameters in a library and control the utility processing by JCL specifications. In which case, modifications or overrides of JCL will need to be made to restart the utility. For example,

Before modification:

```
//SYSIPT DD DSN=IDMS.UTILITY.CNTL(OFFLDJNL),DISP=SHR
```

After modification:

```
//SYSIPT DD DSN=IDMS.UTILITY.CNTL(RSTRAJNL),DISP=SHR
```

2. Recreate the failed journal file - It is likely that the CV (of which the failed disk journal file belongs to) is still active, and would have to be brought down before the failed file can be reallocated and copy over with its counterpart. If the CV is active, and depending on criticality of the online processing and the length of delay that is acceptable, the user can execute the CV journal recovery procedure instead of this procedure. If not, execute this step and continue.
3. Restart the utility - Restart IDMSAJNL with the adjusted parameters and condense both the primary and the duplex disk journal files as normal archive journal execution. Unless the I/O error occurred at the beginning (on the journal header), the archives produced by the prior execution of the utility can be JFIXed and used. In this case, additional JCL would be made in Step 1 to dummy out the archives for this run of IDMSAJNL.
4. Restore the input parameter/JCL - Reverse the JCL modification done in Step 1 to resume normal operation with the next execution of IDMSAJNL.

IDMSAJNL Archive Recovery Procedure

It is important to keep in mind that the proper identification of the entity in error must be made prior to the execution of the recovery procedure. The IDMSAJNL archive recovery procedure is made of several steps, each of which is described here:

SECTION 3: RECOVERY OPERATION

IDMSAJNL ARCHIVE RECOVERY PROCEDURE

- Modify the input parameters/JCL
- Restart the utility
- Restore the input parameters/JCL

The above three steps are the same as Step 1, Step 3 (with the optional JCL changes eliminated) and Step 4 of the IDMSAJNL journal recovery procedure. This procedure is simpler because the entity in error is less critical and is being created.

MESSAGES

DB/DYNAM generates the following messages:

DYNAM-101I Successful dynamic allocation for dd *ddname*

Explanation: The indicated processing request was completed successfully. The field in the message text is:

ddname This field contains the name of the dd for which the request was performed.

DYNAM-102I Successful close for dd *ddname*

Explanation: The indicated processing request was completed successfully. The field in the message text is:

ddname This field contains the name of the dd for which the request was performed.

DYNAM-103I Successful dynamic deallocation for dd *ddname*

Explanation: The indicated processing request was completed successfully. The field in the message text is:

ddname This field contains the name of the dd for which the request was performed.

DYNAM-104I Successful swap for dd *ddname*

Explanation: The swap function was successfully executed. The field in the message text is:

ddname This field contains the name of the dd for which the request was performed.

DYNAM-105I Successful open for dd *ddname*

Explanation: The open function was successfully executed. The field in the message text is:

ddname This field contains the name of the dd for which the request was performed.

SECTION 4: MESSAGES AND CODES

DYNAM-106I Bypass duplex for dd *ddname*

Explanation: The duplex bypass function was successfully executed. The field in the message text is:

ddname This field contains the name of the dd for which the request was performed.

DYNAM-107I Resume duplex for dd *ddname*

Explanation: The duplex process has been activated for the indicated file. The field in the message text is:

ddname This field contains the name of the dd for which the request was performed.

DYNAM-108I Successful start of duplex queuing

Explanation: The start queue command was successfully processed, TDS will now operate under the duplex queuing mode.

DYNAM-109I Successful start of duplex synchronization

Explanation: The start sync command was successfully processed; TDS will now operate under the duplex resynchronization mode.

DYNAM-201E One or more required fields missing or invalid

Explanation: The request was invalid because at least one required keyword was missing or invalid. Correct the request and enter it again.

DYNAM-202E Keyword *kywd* specified twice

Explanation: An error was detected while parsing the input request. The named keyword appeared more than once in the request. The field in the message text is:

kywd keyword. This field names the keyword that was specified twice.

DYNAM-203E Keyword *kywd* conflicts with a preceding KEYWORD

Explanation: An error was detected while parsing the input request. Two mutually exclusive keywords were specified. The field in the message text is:

kywd keyword. This field names the second of the two mutually exclusive keywords.

SECTION 4: MESSAGES AND CODES

DYNAM-204E Expected a value for keyword *kywd* but found end of input

Explanation: DB/DYNAM detected an unexpected end-of-input when searching for the value associated with the last keyword of the input request. The field in the message text is:

kywd keyword. This field names the keyword whose associated value was not found.

DYNAM-205E Input token *it* not recognized

Explanation: An unrecognized input token was detected while parsing the input request. The field in the message text is:

it input token. This field names the unrecognized token.

DYNAM-206E Input token *it* too long

Explanation: An input token was detected whose length was greater than that permitted. The field in the message text is:

it input token. This field names the token that was too long.

DYNAM-301E Close error, return code *rc* received

Explanation: An error occurred during the execution of a CLOSE macro. The field in the message text is:

rc return code. This field indicates the reason for the error.

DYNAM-302E Close error, specified ddname not found in CV

Explanation: The close request was terminated because the ddname specified in the request was not found in the DMCL (for an IDMS/R database) or in the DPLXTABL table (for a DB/DUPLEX duplex database). Close requests for non-IDMS/R files are invalid.

DYNAM-303E Close error, dd *ddname* not open

Explanation: The close request was terminated because the file whose ddname appears in the message was not open. The field in the message text is:

ddname ddname. This field indicates the ddname of the file which was not open.

SECTION 4: MESSAGES AND CODES

DYNAM-304E Area *an* in *um* mode, close not performed

Explanation: The close request was terminated because one of the areas related to the file being closed was not offline. The fields in the message text are:

an area name. This field names the area that was not offline.

um usage mode. This field names the usage mode of the area, *an*.

DYNAM-311E Duplex file unavailable for *op* operation

Explanation: A duplex file is not available for the specified operation. The field in the message text is:

op operation. The field indicates the requested dynamic duplex function.

DYNAM-312E *op* error, specified *ddname* not found in CV

Explanation: The requested function was terminated because the *ddname* specified in the request was not found in the DMCL (for an IDMS/R database) or in the DPLXTABL table (for a DB/DUPLEX duplex database). Operation requests for non-IDMS/R files are invalid. The field in the message text is:

op operation. The field indicates the requested dynamic duplex function.

DYNAM-313E *op* error, *dd* *ddname* not open

Explanation: The specified operation was not executed due to the close file status. The fields in the message text are:

op operation. The field indicates the requested dynamic duplex function.

ddname *ddname*. This field indicates the *ddname* of the file which was not open.

DYNAM-314E Area *an* in *um* mode, *op* not performed

Explanation: The operation request was terminated because one of the areas related to the file being operated on was not offline. The fields in the message text are:

an area name. This field names the area that was not offline.

um usage mode. This field names the usage mode of the area, *an*.

SECTION 4: MESSAGES AND CODES

op operation. The field indicates the requested dynamic duplex function.

DYNAM-315E File *ddname* have been swapped, *op* not performed

Explanation: The request was terminated due to the swap status of the file. The fields in the message text are:

ddname *ddname*. This field indicates the *ddname* of the file which was not open.

op operation. The field indicates the requested dynamic duplex function.

DYNAM-321E Open error, return code *rc* received

Explanation: An error occurred during the execution of an OPEN macro. The field in the message text is:

rc return code. This field indicates the reason for the error.

DYNAM-322E Open error, specified *ddname* not found in CV

Explanation: The open request was terminated because the *ddname* specified in the request was not found in the DMCL (for an IDMS/R database) or in the DPLXTABL table (for a DB/DUPLEX duplex database). Open requests for non-IDMS/R files are invalid.

DYNAM-323E Open error, *dd* *ddname* already opened

Explanation: The open request was terminated due to the specified file status indicating it is already opened. The field in the message text is:

ddname *ddname*. This field indicates the *ddname* of the file which was not open.

DYNAM-401W Warning, specified *ddname* not found in CV

Explanation: A possible error was detected. The *ddname* specified in the input request could not be found in the DMCL (for IDMS/R databases and IDMS/R files) or in the DPLXTABL table (for duplex databases).

DYNAM-402E File already open, allocation not done

Explanation: The allocation request was terminated because IDMS-CV/DC or DB/DUPLEX indicated that the file associated with the request was already open.

SECTION 4: MESSAGES AND CODES

DYNAM-403E Invalid parameter *ip* detected

Explanation: While processing the dynamic allocation/deallocation request, a field was detected to have invalid contents. The field in the message text is:

ip invalid parameter. This field displays the contents of the invalid parameter.

DYNAM-404E Request denied by user installation routine

Explanation: The user installation validation routine denied the dynamic allocation/deallocation request. The routine uses exit IEFDB401. For more information, contact your system programmer.

DYNAM-405E JOBLIB/STEPLIB/JOBCAT/STEPDAT incorrectly specified as ddname

Explanation: The input request incorrectly specified one of the ddnames contained in the message. The dynamic allocation/deallocation request was rejected.

DYNAM-406E Authorized function requested by unauthorized user

Explanation: The dynamic allocation/deallocation request tried to perform an authorized function, but the requesting program was not authorized to perform it. The most common cause of this error is an attempt to mount a device or to use an offline device. To accomplish the authorized function, the requesting program must meet at least one of the following criteria:

- it must have a system storage protection key (0-7)
- it must be in supervisor state
- it must be APF authorized

DYNAM-407E Specified ddname unavailable

Explanation: The dynamic allocation request was rejected because the ddname specified in the input request was unavailable for use. This error is generated because the specified ddname is already in use for another dataset.

DYNAM-408E Specified ddname or dsname associated with an open dataset

Explanation: The dynamic deallocation request was rejected because the ddname specified in the input request was not closed.

SECTION 4: MESSAGES AND CODES

DYNAM-409E Specified ddname not found

Explanation: The dynamic deallocation request was rejected because the ddname specified in the input request could not be found among all the dd's currently allocated to IDMS-CV/DC.

DYNAM-410E Specified dataset not in catalog

Explanation: The dynamic allocation request was rejected because the dsname specified in the input request could not be located in any of the catalogs accessible to IDMS-CV/DC.

DYNAM-411E Specified dataset unavailable

Explanation: The dynamic allocation request was rejected because the dataset specified in the input request was not available for allocation. This is usually caused by another job having exclusive control of the dataset.

DYNAM-412E Requested volume unavailable

Explanation: The dynamic allocation request was rejected because the dataset specified in the input request was on a volume that was not available to IDMS-CV/DC.

DYNAM-499E Processing error, return code *rc*, error code *xec*, INFO CODE *Xic*

Explanation: An error occurred during the execution of a dynamic allocation or dynamic deallocation request. The fields in the message text are:

rc return code. This field indicates the general cause of the error. The return codes and their meanings are listed below.

ec error code. This field indicates the specific cause of the error. The error codes and their meanings are listed after the return codes.

ic information code. This field appears only for certain error codes and it supplies additional information for those error codes. The information codes and their meanings are listed after the error codes.

SECTION 4: MESSAGES AND CODES

The values of *rc* and their meanings are as follows:

Return Code	Meaning
04	An error resulted from the current environment, the unavailability of a system resource, or a system routine failure.
08	Your installation validation routine denied this request. Contact your system programmer for more information.
0C	The error is due to an invalid parameter list.

The values of *ec* and their meanings are as follows:

Error Code	Meaning
0204	Real storage unavailable.
0208	Reserved.
020C	Request for exclusive use of a shared dataset cannot be honored. ¹
0210	Request dataset unavailable. The dataset is allocated to another job and its usage attribute conflicts with this request. ¹
0214	Unit(s) not available; or, if allocating an internal reader, all defined internal readers are already allocated. ¹
0218	Specified volume or an acceptable volume is not mounted, and user does not have volume mounting authorization. ¹
021C	Unit name specified is undefined.
0220	Requested volume not available. ²
0224	Eligible device types do not contain enough units. ¹
0228	Specified volume or unit in use by system.
022C	Volume mounted on ineligible permanently resident or reserved unit.
0230	Permanently resident or reserved volume on required unit.

¹ The conditions that cause these return codes are detected by MVS/XA or JES3.

² For MSS requests, the MSSC reason code for this failing job step is contained in message IEF710I on the hardcopy log. An explanation of the MSSC reason code is contained in the IBM manual Mass Storage System Extensions Messages. For non-MSS requests, this code is accompanied by message IEF485I. It may result from a JES3 failure because of a busy or unavailable situation.

SECTION 4: MESSAGES AND CODES

Error Code	Meaning
0234	More than one device required for a request specifying a specific unit.
0238	Space unavailable in task input output table (TIOT).
023C	Required catalog not mounted, and user does not have volume mounting authorization.
0240	Requested device is a console.
0244	Telecommunication device not accessible.
0248	MSS virtual volume cannot be mounted.
024C	Operating-system-managed resource was unavailable to the subsystem. ³
0250	Subsystem resource not available. ¹
0254	The TIOT resource is currently unavailable and the user requested conditional ENQ on the resource.
0258	There were not a sufficient number of non-restricted units to satisfy the request, or JES3 selected a JES3-managed restricted unit to satisfy the request. Restricted units are described in the IBM manual SPL: System Generation Reference.
025C	Requested device is boxed and cannot be accessed, as a result of an I/O error condition or the operator issuing a VARY X, OFFLINE, FORCE command.
0260	Unit does not meet specified status requirements.
0264	Invalid request due to current unit status.
0268	Tape device is broken.
0304-0338	Assigned by DAIR. (See the IBM manual TSO Guide to Writing to Terminal 0338 Monitor Program or a Command Processor.)
033C-0354	Reserved.
0358	Overriding disposition of DELETE invalid for dataset allocated as SHR ²

¹ The information code field contains a subsystem-defined value to further describe the error. This value is documented in publications associated with the particular subsystem.

² The information code field contains 0004 if the requested function was performed, although an error occurred, as the error code indicates.

SECTION 4: MESSAGES AND CODES

Error Code	Meaning
035C	Invalid PARM specified in text unit. ¹
0360	Invalid KEY specified in text unit. ¹
0368	Authorized function requested by unauthorized user.
036C	Invalid parameter list format.
0364	JOBLIB/STEPLIB/JOBCAT/STEP CAT specified as ddname, or associated with specified dsname. ²
0370	Reserved.
0374	Invalid # specified in text unit. ¹
0378	Duplicate KEY specified in text unit. ¹
037C	Invalid LEN specified in text unit. ¹
0380	Mutually exclusive KEY specified. Two keys that cannot be used together were used in the request. ¹
0384	Mutually inclusive KEY not specified. One key was used; two should have been used. ¹
0388	Required key not specified.
038C	Duplicate ddnames specified.
0390	GDG group name specified with relative generation number exceeds 35 characters.
0394	Status and relative generation number are incompatible.
0398	Volume sequence number exceeds the number of volumes.
039C	Device type and volume are incompatible.
03A0	Subsystem detected an invalid parameter. ³
03A4	Unable to protect dataset/volume because of conflicting keyword specification.

¹ The information code contains the value of the key that caused the error.

² The information code field contains 0004 if the requested function was performed, although an error occurred, as the error code indicates.

³ The information code field contains a subsystem-defined value to further describe the error. This value is documented in publications associated with the particular subsystem.

SECTION 4: MESSAGES AND CODES

Error Code	Meaning
0404-040C	Reserved.
0410	Specified ddname unavailable.
0414-041C	Reserved.
0420	Specified ddname or dsname associated with an open dataset. ¹
0424	Deconcatenation would result in duplicate ddnames. ¹
0428-0430	Reserved.
0434	DDname specified in ddname allocation request is associated with a convertible or non-permanently allocated resource.
0438	Specified ddname not found.
043C	The system could not deallocate enough of the resources being held in anticipation of reuse to meet the control limit.
0440	Specified dsname not found.
0444	Relative entry number specified in information retrieval request not found.
0448	Request for a new dataset failed; the dataset already exists.
044C	Request was made for a dataset that has a disposition of delete; this request cannot be honored because the dataset may be deleted at any time.
0450	Request would cause the limit of 1635 concurrent allocations to be exceeded.
0454	DDname in DCB reference not found.
0458	Dsname in DCB reference or volume reference is a GDG group name.
045C	Specified dsname to be deallocated is a member of a permanently-concatenated group. ¹
0460	Specified dsname or member to be deallocated is not associated with specified ddname.

¹ The information code field contains 0004 if the requested function was performed, although an error occurred, as the error code indicates.

SECTION 4: MESSAGES AND CODES

Error Code	Meaning
0464	Specified dsname to be deallocated is a private catalog. ¹
0468	Error while allocating or opening a private catalog.
046C	Remote work station not defined to job entry subsystem.
0470	User unauthorized for job entry subsystem request.
0474	Error while attempting to select optimum device.
0478	Unable to process job entry subsystem request.
047C	Unable to establish ESTAE environment.
0480	The number of units needed to satisfy the request exceeds the limit.
0484	Request denied by operator.
0488	GDG pattern DSCB not mounted.
048C	GDG pattern DSCB not found.
0490	Error changing allocation assignments.
0494	Error processing OS CVOL.
0498	MSS virtual volume not accessible. ²
049C	MSS virtual volume not defined. ³
04A0	Specified MSVGP name not defined. ⁴
04A4	Subsystem request in error. ⁵
04A8	Subsystem does not support allocation via key DALSSNM.

¹ The information code field contains 0004 if the requested function was performed, although an error occurred, as the error code indicates.

² The MSSC reason code for this failing job step is contained in message IEF710I on the hardcopy log. An explanation of the MSSC reason code is contained in the IBM manual Mass Storage System Extensions Messages.

³ This code corresponds to MSSC reason code X'007', which is explained in the IBM manual Mass Storage System Extensions Messages.

⁴ This code corresponds to MSSC reason code X'207', which is explained in the IBM manual Mass Storage System Extensions Messages.

⁵ The information code contains a subsystem-defined value to further describe the error. This value is documented in publications associated with the particular subsystem.

SECTION 4: MESSAGES AND CODES

Error Code	Meaning
04AC	Subsystem is not operational.
04B0	Subsystem does not exist.
04B4	Protect request not processed; RACF not in system or not active.
04B8	MSS not initialized for allocation.
04BC	MSS volume select error. ¹
04C0	Protect request failed; user not defined to RACF.
04C4	The last request was for a VOL=REF to a dsname or DCB=dsname that exceeded the maximum allowable dsname referbacks. (A maximum of 972 backward references are allowed if the dataset names are 44 characters in length.)
04C8	A non-zero return code was set from either common allocation or JFCB housekeeping; however, the SIOT reason code (SIOTRSNC) was not set. This problem might result from installation modification of the eligible device table (EDT).
04CC	Invalid reference to an OUTPUT JCL statement.
17zz	LOCATE error. (Note: Hexadecimal '08', '18', and '2C' are the only expected LOCATE return codes. 'FF' is returned as the value of zz if an unexpected return code is returned by LOCATE.) ²
27zz	Reserved.
37zz	Reserved.
47zz	DADSM allocation error. ³
57zz	CATALOG error. ²

Note: The failing system routine returns the code represented by "zz".

¹ The information code field contains the MSSC reason code. An explanation of the MSSC reason code is contained in the IBM manual Mass Storage System Extensions Messages

² For detailed descriptions of the LOCATE, CATALOG and OBTAIN return codes, see the IBM manual Data Management.

³ The information code field might contain a value that further describes the error. An explanation of this value is contained in the allocation message corresponding to the error code. See the section on reason codes in the allocation/deallocation component of the IBM System Logic Library for the message number associated with the error and the IBM Message Library: System Messages for the message itself.

SECTION 4: MESSAGES AND CODES

Error Code	Meaning
67zz	OBTAIN error. ¹
7700	Subsystem error. ²
7704	A subsystem interface system error occurred while processing key DALSSNM.
8700	Scheduler JCL facility error.

The values of *ic* and their meanings are as follows:

Info Code	Meaning
0004	Reserved.
0008	Overriding disposition ignored for one of the following reasons: <ul style="list-style-type: none">• Dataset was originally allocated with a disposition of PASS• Dataset is a non-subsystem dataset that has a system-generated name; you cannot override disposition on this type of dataset• Dataset is a VSAM dataset. In these cases, the dataset is deallocated using the disposition specified when the request was allocated.
000C-001C	Reserved.
002n	The dataset was successfully deallocated but processing of the requested CATLG or UNCATLG disposition was unsuccessful. The digit "n" is a code representing the reason for the failure. The possible codes and their meanings are: Code Meaning 1 A control volume is required; a utility program must be used to catalog the dataset.

Note: The failing system routine returns the code represented by "zz".

¹ For detailed descriptions of the LOCATE, CATALOG and OBTAIN return codes, see the IBM manual Data Management.

² The information code field contains a subsystem-defined value to further describe the error. This value is documented in publications associated with the particular subsystem.

SECTION 4: MESSAGES AND CODES

Code Meaning

- 1 A control volume is required; a utility program must be used to catalog the dataset.
- 2 The dataset to be cataloged is already cataloged; or the dataset to be uncataloged could not be located; or no change was made to the volume serial list of a dataset with a disposition of CATLG.
- 3 The specified index does not exist.
- 4 The dataset could not be cataloged because the space was not available in the catalog.
- 5 Not enough storage was available to perform the specified cataloging.
- 6 The dataset to be cataloged in a generation index is improperly named.
- 7 The dataset to be cataloged has not been opened; no density information is available (for dual density tape requests only).
- 8 Reserved.
- 9 An uncorrectable I/O error occurred in reading or writing the catalog.

003n The dataset was successfully deallocated but processing of the requested DELETE disposition was unsuccessful. The digit "n" is a code representing the reason for the failure. The possible codes and their meanings are:

Code Meaning

- 1 The expiration date has not occurred.
- 2 Reserved.
- 3 Reserved.
- 4 No device was available for mounting the volume during deletion.
- 5 Not enough storage was available to perform the specified deletion.
- 6 Either no volumes were mounted or volumes that were mounted could not be demounted to permit the remaining volumes to be mounted.

SECTION 4: MESSAGES AND CODES

Code Meaning

- 8 The SCRATCH routine returned an error code. If the user's JOB statement requested allocation/ termination messages, message IEF283I appears in the SYSOUT listing. This message lists the volume serial numbers of the datasets that were not deleted; following each number is a code that explains why each dataset was not deleted.

DYNAM-501W Warning, Duplex Queuing is currently active

Explanation: A start queue command was issued in the duplex queuing mode.

DYNAM-502W Warning, Duplex Synchronization is currently active

Explanation: A start sync command was issued in the duplex synchronization mode.

DYNAM-511E Duplex Synchronization not activated, Duplex Queuing is not active

Explanation: Online duplex synchronization can only be started in the duplex queuing mode.

DYNAM-512E Duplex Queuing not activated, not all affected area(s) are offline

Explanation: The start queue command was terminated due to the area status of the files being duplexed. Duplex queuing can only be started when all affected areas are offline.

DYNAM-601W Warning, specified ddname already being bypassed

Explanation: A bypass command was issued against a file that is not being duplexed.

DYNAM-701W Warning, specified ddname already being duplexed

Explanation: A duplex command was issued against a file that is being duplexed.

SECTION 4: MESSAGES AND CODES

DB/INFO generates the following messages:

DBINFO-201E One or more required fields missing or invalid

Explanation: The request was invalid because at least one required keyword was missing or invalid. Correct the request and enter it again.

DBINFO-202E Keyword *kywd* specified twice

Explanation: An error was detected while parsing the input request. The named keyword appeared more than once in the request. The field in the message text is:

kywd keyword. This field names the keyword that was specified twice.

DBINFO-203E Keyword *kywd* conflicts with a preceding KEYWORD

Explanation: An error was detected while parsing the input request. Two mutually exclusive keywords were specified. The field in the message text is:

kywd keyword. This field names the second of the two mutually exclusive keywords.

DBINFO-204E Expected a value for keyword *kywd* but found end of input

Explanation: DB/INFO detected an unexpected end-of-input when searching for the value associated with the last keyword of the input request. The field in the message text is:

kywd keyword. This field names the keyword whose associated value was not found.

DBINFO-205E Input token *it* not recognized

Explanation: An unrecognized input token was detected while parsing the input request. The field in the message text is:

it input token. This field names the unrecognized token.

DBINFO-206E Input token *it* too long

Explanation: An input token was detected whose length was greater than that permitted. The field in the message text is:

it input token. This field names the token that was too long.

SECTION 4: MESSAGES AND CODES

Technetron Duplex Subsystem generates the following messages:

TUDSMAIN - No entry found in DPLXTABL for DDname=dddddddd

Explanation: The requested function was terminated since no translation was found in the DPLXTABL table for the ddname *dddddddd* specified. Regenerate the DPLXTABL table to include the specified DDname. The field in the message text is:

dddddddd ddname. This field contains the ddname of the primary file (the one in the DMCL) for which the request was performed.

TUDSMAIN - Warning, duplex file dddddddd is not open

Explanation: This message is generated when IDMS begins updating the primary file, which corresponds to the duplex file in the message, and the duplex file is defined in the DPLXTABL table but not defined or is dummied out in the execution JCL. This message serves as a reminder that duplexing is not being performed for the named duplex file. The field in the message text is:

dddddddd ddname. This field contains the ddname of the duplex file which is not open.

xxxxxxx - Warning, duplex file dddddddd is not available due to a previous I/O error

Explanation: One of the TDS online component's modules, represented by *xxxxxxx*, has encountered a duplex file with I/O problems that has been varied offline. The fields in the message text are:

dddddddd ddname. This field contains the ddname of the file that experienced the I/O problem.

xxxxxxx This field contains the name of the module (TUDSDBJN, TUDSMAIN, TUDSOPTZ) which has encountered a duplex file with an I/O problem.

TUDSMAIN - Duplex (ddddddd) blk size nnnnn does not equal primary (pppppppp) blk size #####

Explanation: The duplex file block size does not match that of the primary. This message appears when *nnnnn* does not equal to *#####*. The fields in the message text are:

ddddddd This field contains the ddname of the duplex file.

pppppppp This field contains the ddname of the primary file.

SECTION 4: MESSAGES AND CODES

nnnnn This field contains the block size of the duplex file

This field contains the block size of the primary.

TUSYNAD-001E ANALYSIS *jjjjjjj,ssssssss,aaa,ddddddd,ooooo,msg,am*

Explanation: The I/O error handler has been invoked to analyze an I/O problem. This information was pass from the operating system. The fields in the message text are:

jjjjjjj Job Name

ssssssss Step Name

aaa Device Address

ddddddd DDname

ooooo I/O Operation

msg Brief explanation of I/O problem

am Access Method

TUSYNAD-002I Snap dump written to DD=TUSNAP, snap id *sss*

Explanation: A snap has been taken successfully. The dump is written to the file or sysout with the DDname TUSNAP. The field in the message text is:

sss This field is an eye-catcher for differentiating multiple snap dumps.

TUSYNAD-003I I/O error detected, duplex file *ddddddd* **closed**

Explanation: The duplex I/O error handler has detected an I/O error. The field in the message text is:

ddddddd This field contains the ddname of the affected duplex file which was closed.

SECTION 4: MESSAGES AND CODES

Technetron Dynamic Facility generates the following IDMS-DC abend codes:

DYN1/INF1

DB/DYNAM or DB/INFO issues this abend code whenever it detects an internal problem. The contents of the halfword pointed to by register 3 represent the cause of the error. Its value and their meaning follow:

- 0001** An unknown area status was discovered while DB/DYNAM or DB/INFO validated the usage mode of all areas within the dataset specified in the issued command.

DYN2/INF2

TDF terminal I/O module was unable to perform an IDMS-DC line mode command successfully.

DYN3/INF3

TDF write log module received a non-zero return code after issuing a #WTL macro. For a detailed description of the return code, refer to IDMS-DB/DC DML Reference-Assembler.

DYN4/INF4

TDF was unable to pass control from one module to another. Check the SYSGEN definition for DB/DYNAM and DB/INFO. Make sure all their modules (programs) are defined to the system (central version) which the TDF task was invoked.

DYN5/INF5

TDF was unable to locate the central control block for TDS. If TDS is installed in the system, storage overlayed may have occurred, contact vendor for further instruction.

INF6

The DPLTABL is not at the release level to operate with TDF, check the DPLXTABL definition.

SECTION 4: MESSAGES AND CODES

Technetron Duplex Subsystem abends with the following system codes:

U1001

The duplex I/O error routine has detected a problem on one of the duplex file generated with MODE=ACTIVE in the duplex table (DPLXTABL). Check other messages to determine the problem file.

U1002

When the duplex I/O error handler was unable to snap dump the related information of a duplex file experiencing I/O difficulties, this abend is issued.

U1003

The duplex I/O error handler is unable to perform the SYNAD analysis function. Contact vendor for further instructions.

U4001

This is generated for abnormal conditions encountered by the TDS online component. Check for other messages, check that the duplicate datasets have been allocated and initialized correctly, and check that the DPLXTABL table has been correctly generated and link-edited properly.

U4002

The duplex buffer that is used for parallel processing of the duplex files may have been corrupted. Contact vendor for further instructions.

INSTALLATION

Technetron software products are easy to install. The software modules have been pre-assembled, and copied onto a single reel 9-track EBCDIC tape. The installation procedure is to 1) unload the tape's content, then 2) modify and execute the unloaded sample jobs.

Depending on the software products to be install, the installer may be required to modify and execute more than just one sample job. The second step of the installation procedure requires the installer to identify the Technetron Licensed Products to be installed in order to select the proper sample job(s).

The installation for Technetron CA-IDMS related software has been separated into two groups, one for the duplex products and another for the dynamic facility. After identifying and selecting the licensed products to be installed, determine which group the selected products are a part of, and follow the instruction in the appropriate section to complete the installation.

TAPE AND CONTENT

The install tape is an IBM Standard Label (SL) formatted tape. There are two sets of two Partition DataSets (PDSs), source and object codes on the tape. The four files on the tape were created by the IBM IEHMOVE utility which "sequentialized" PDS libraries into QSAM files. The first two files are:

1. Filename: TU.MVS.FILE1
Format: Unloaded PDS in IEHMOVE format.
Content: Source entities needed for the installation procedure including duplex table definition macros.

2. Filename: TU.MVS.FILE2
Format: Unloaded PDS in IEHMOVE format.
Content: Pre-assembled XPO object modules.

The first two files are repeated to create the next two files, so file-3 and file-4 on the install tape matches exactly as file-1 and file-2. The two sets of files are provided in case of a media problem on any of the files. Each file is a library of objects that are needed to complete the software installation.

The first file and the third file, TU.MVS.FILE1, are the source PDS that contains macros, control cards, sample install jobs, and sample source programs. The members in the library are:

<u>Name</u>	<u>Description</u>
#DPLXEND	Duplex table footer definition macro
#DPLXHR	Duplex table header definition macro
#DPLXMAP	Duplex table entry mapping definition macro

**SECTION 5: INSTALLATION
TAPE AND CONTENT**

<u>Name</u>	<u>Description</u>
CTLCAJDX	ARCHIVE/DUPLEX linkage editor control statements
CTLCDBDX	DB/DUPLEX linkage editor control statements
CTLCDXUX	TDS user exits linkage editor control statements
CTLCDYNM	DB/DYNAM linkage editor control statements
CTLCINFO	DB/INFO linkage editor control statements
CTLCJNDX	JOURNAL/DUPLEX linkage editor control statements
CTLCTDS	TDS linkage editor control statements
CUSTDYNM	DB/DYNAM and DB/INFO task and program sysgen definition and IDMS-DC message update sample job
DPLXTAB1	Sample DPLXTABL generation job for JOURNAL/DUPLEX using the duplex table definition macros
DPLXTAB2	Sample DPLXTABL generation job for JOURNAL/DUPLEX and DB/DUPLEX using the duplex table definition macros
INSTDPLX	Sample duplex product (JOURNAL/DUPLEX) installation job
INSTDXUX	TDS user exits sample installation JCL
INSTDYN2	Sample installation job in creating a duplicate DB/DYNAM control module that does not perform terminal I/O
INSTDYNM	DB/DYNAM and DB/INFO sample installation JCL
MSGSDYNM	DB/DYNAM data dictionary messages
MSGSINFO	DB/INFO data dictionary messages
SYSGDYNM	DB/DYNAM sample program and task definition sysgen control statements
SYSGINFO	DB/INFO sample program and task definition sysgen control statements
XMP1DYNM	DB/DYNAM sample IDMS-DC COBOL program utilizing DYNAM task code
XMP2DYNM	DB/DYNAM sample IDMS-DC COBOL program linking to DB/DYNAM control module directly
XMP3DYNM	DB/DYNAM sample IDMS-DC Assembler program calling DB/DYNAM control module directly

The second and the fourth file, TU.MVS.FILE2, contains the executable modules (object decks) of XPO, they are:

<u>Name</u>	<u>Description</u>
TUDFDDYN	DB/DYNAM control module
TUDFDPXN	DB/INFO duplex information module
TUDFDXOP	DB/DYNAM duplex operation module
TUDFINFO	DB/INFO control module
TUDFLODN	DB/DYNAM load module information module
TUDFSQSN	DB/DYNAM duplex queue/synchro activation module
TUDFSYOP	DB/DYNAM system operation module
TUDFUX13	DUPLEX/SYNCHRO duplex queue CV shutdown module
TUDSAJDX	ARCHIVE/DUPLEX utility module
TUDSBSNC	TDS batch duplex queue synchronization module

SECTION 5: INSTALLATION UNLOAD TAPE

<u>Name</u>	<u>Description</u>
TUDSDBJN	TDS online component duplex I/O module
TUDSDXSN	TDS online duplex queue/synchronization module
TUDSJ2DX	JOURNAL/DUPLEX sequential journal duplex module
TUDSMAIN	TDS online component duplex control module
TUDSOPTZ	TDS online component optimize module
TUDSUTL1	DUPLEX/SYNCHRO utility module
TUSYNAD	TDS online component I/O error handler module

The objects were created by Assembler H (IEV90). The XPO component that a module belongs to is identified by its first four characters, TUDS for TDS, and TUDF for TDF, the exception is TUSYNAD which is the TDS's I/O error handler.

STEP 1 - UNLOAD TAPE

The first step of the installation is to create an unload job similar to the following sample JCLs. The unload job should offload two files from the installation tape, rename and catalog them.

```

c                               1                               7
c 1.....6.....2
//UNLOAD1 EXEC PGM=IEHMOVE,REGION=512K
//SYSPRINT DD SYSOUT=*
//TAPEDD DD DISP=OLD,UNIT=tape,VOL=SER=vvvvvv,
// LABEL=(,SL),DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//DISKDD DD VOL=SER=dddddd,UNIT=disk,DISP=OLD
//SYSUT1 DD VOL=SER=ssssss,UNIT=disk,DISP=OLD
//SYSIN DD *
COPY PDS=TU.MVS.FILE1,FROM=tape=(vvvvvv,1), *
FROMDD=TAPEDD,TO=disk=dddddd,CATLG, *
RENAME=TU.TEST.SRCLIB
COPY PDS=TU.MVS.FILE2,FROM=tape=(vvvvvv,2), *
FROMDD=TAPEDD,TO=disk=dddddd,CATLG, *
RENAME=TU.TEST.OBJLIB

```

Explanations:

tape = Unit parameter for TAPE drives
 disk = Unit parameter for DISK volume
 vvvvvv = Volume serial number of the installation tape
 ddddddd = Volume serial number of the disk that will receive the offloaded files
 ssssss = Volume serial number of a scratch disk pack

Execution of the above JCL will unload the first two files, TU.MVS.FILE1 and TU.MVS.FILE2, from the tape and create two partitioned datasets named TU.TEST.SRCLIB and TU.TEST.OBJLIB. Change the sequence number on the FROM=tape=(vvvvvv,#) of the COPY control statement to unload file-3 or file-4 if there is problem unloading the first two files.

SECTION 5: INSTALLATION MODIFY AND EXECUTE - INSTDPLX

The content of the unloaded files should correspond to the library member list which was specified on the previous pages. The integrated installation tape will no longer be needed; from this point on, the installation procedure will proceed with the unloaded datasets.

Please follow your installation naming standard as you may be unable to catalog datasets with the high qualifier of TU. The sample JCLs supplied with the tape use these names as catalog partition datasets, if the library names are changed during the unload, the installer should note the changes and apply them in the subsequent step of the installation.

STEP 2 - MODIFY AND EXECUTE

The second step of the install procedure is to modify the sample JCLs for the products which are to be installed. The products are categorized as one of the duplex products or a part of the dynamic facility. The following lists the products by their grouping:

Duplex Products: ARCHIVE/DUPLEX
 DB/DUPLEX
 DUPLEX/SYNCHRO
 JOURNAL/DUPLEX

Dynamic Products: DB/INFO
 DB/DYNAM

INSTDPLX

INSTDPLX is a member of source library that should be used to install the duplex products. Modify this member according to your site standards on items such as jobname, job class and job accounting information, and also modify the library names if they were changed during the Step 1 execution.

INSTDPLX allocates a load module library and link-edit the duplex object modules into the new load library. The control of link-editing which duplex modules (License Products) is in the SYSLIN DD statement of the sample job's second step (LINK).

The linkage editor control statements have been logically group and supplied as members in the source library. Please refer to the members listed on page 5-2 for the different configurations of duplex software, select and substitute the appropriate ones for your installation.

For individual duplex product installation, change the default member name of **CTLCJNDX** (for JOURNAL/DUPLEX) in the SYSLIN DD statement to **CTCLAJDX** for ARCHIVE/DUPLEX and to **CTCLDBDX** for DB/DUPLEX. For XPO and TDS users, replace the default of CTCLJNDX with **CTLCTDS**.

SECTION 5: INSTALLATION MODIFY AND EXECUTE - INSTDYNM

Please note that the control statement members were setup to work with the sample installation jobs. The control statement INCLUDE parameters match the DDnames of the sample jobs, both should be modified to the same name if either one was modified.

After modifying the sample JCL, submit the INSTDPLX for execution. Your duplex software installation procedure is complete as of the successful execution the job. A duplex table (DPLXTABL) is a required operating entity of the TDS online component. Examples of duplex table generation have been provided in the source library. Modify the appropriate member and submit it for execution. The description of duplex table generation and other duplex related operation can be found in Section 2, System Operation, please refer to that section for customization and integration procedures.

This section is for XPO sites that will implement the automatic duplex queue resynchronization feature during IDMS shutdown. The automatic shutdown resync module is to be integrated with IDMS as one of the system-provided Numbered Exits. Follow the installation instruction for the Numbered Exits in the CA-IDMS System Operation manual to install the duplex queue auto-shutdown-resync feature.

In general, modify the IDMS-DC Numbered User Exit definition module, RHDCUXIT, by replacing Numbered Exit 13 line with the following:

```

c          1          7
c 1.....0.....2
           #DEFXIT MODE=SYSTEM, CALL=DC, EP=UX13EP1
```

and edit the corresponding linkage editor control statements to include the following control card:

```

c          7
c 1.....2
           MODE AMODE (ANY) RMODE (24)
           INCLUDE OBJ (TUDFUX13)
```

After the modification, assemble and link-edit the RHDCUXIT module below the 16-meg line to complete the duplex software installation. It is essential that RHDCUXIT resides below the 16-meg line due to the automatic shutdown resync module issuance of BDAM I/Os, which must be in 24-bit mode.

INSTDYNM

The member that will be use to install the dynamic facility is INSTDYNM. It is located in the source library. Modify this member according to your installation standards on items such as jobname, job class and job accounting information, and also modify the library names if they were changed during the Step 1 execution.

SECTION 5: INSTALLATION

MODIFY AND EXECUTE - CUSTDYNM

INSTDYNM allocates a load module library and link-edit the dynamic object modules into the new load library. The control of link-editing which dynamic facility's modules (License Products) is in the SYSLIN DD statement of the sample job's second step (LINK).

INSTDYNM has been supplied to install TDF, both DB/DYNAM and DB/INFO products. For users who are installing only DB/DYNAM, delete the concatenated SYSLIN DD statement which contain the DB/INFO control cards, CTLCINFO. XPO users should consider combining the duplex products and the dynamic products into one load library, if so, delete the first (ALLOCATE) step of INSTDYNM, and modify the LINK step's SYSLMOD DD statement and the site specifics parameters and submit INSTDYNM for execution.

The entire dynamic facility installation is made of two parts, dynamic software installation and system customization. The dynamic software installation is completed after the successful execution of INSTDYNM. The dynamic facility customization is system dependent; it should be done for each Central Version that requires TDF's operation.

CUSTDYNM

The CV customization sample job is provided as member CUSTDYNM in the source library. CUSTDYNM is made of two steps, defining the dynamic products task and program to a particular CV, and adding the corresponding dictionary messages to that CV. The sample job updates CV 99's dictionary. Users should use this job as a template, modify the CV number, user signon, and JCLs, then submit the job to the CV that the dynamic products will run on.

For DB/DYNAM users who did not install the DB/INFO modules in the load library, delete the concatenated SYSIPT DD statements containing the DB/INFO sysgen definition and messages from the SYSGEN and ADDMSGs steps of INSTDYNM. XPO users should modify and execute INSTDYNM with only installation specifics modifications.

The installation of the dynamic facility is now complete; refer to Section 2, System Operation, on TDF usage. Technetron software products are automatically and dynamically integrated with each other, once installed, they will operate based on the presence or absence of other products.

#DPLXEND	2-9	INFO LOADLIB	2-26
#DPLXHR	2-6	DMCL	2-7, 2-10, 2-11, 2-17, 2-23, 3-10, 3-11
#DPLXMAP	2-6 – 2-9	DPLXTABL	1-5, 1-7, 2-5 – 2-7, 2-9 – 2-10, 2-14, 2-19, 2-23, 2-24, 2-25, 2-26
A			
abend	<i>see user abend</i>	dual journaling	1-1, 1-3
abnormal termination	2-5, 2-8, 3-1	Duplex products	5-4
Archive	<i>see journal</i>	Duplex queue	2-1 – 2-5
archive		DUPLEX/SYNCHRO	1-5, 2-2, 2-4, 2-10, 2-12, 5-2, 5-3, 5-4
archive journal	<i>see IDMSAJNL</i>	duplex table	<i>see DPLXTABL</i>
ARCHIVE/DUPLEX	1-5, 1-6, 2-11, 3-10, 5-2, 5-4	Dynamic products	5-4, 5-6
automatic recovery	1-3, 3-5	I	
B			
Batch Component	1-4, 2-10 – 2-12	I/O error	1-3, 1-5, 1-6, 2-6, 2-8, 3-1 – 3-11
C			
Central Version/CV	1-5, 1-6, 1-7, 2-1, 2-2, 2-3, 2-13, 2-14, 2-16, 2-17, 3-1 – 3-5, 3-11	handling	3-1
Codes	<i>see user abend</i>	"hard"	3-1, 3-2
Control module	<i>see duplex modules</i>	recovery	<i>see recovery operation</i>
CUSTDYNM	5-6	I/O error handler	<i>see duplex modules</i>
D			
database	1-1 - 1-7, 2-1 – 2-10, 2-12 – 2-26, 3-1 – 3-3, 3-5 – 3-10	IDMSAJNL	1-5, 1-6, 2-11, 2-15, 2-16, 3-10 – 3-12
DB/DUPLEX	1-3, 1-4, 1-6, 2-1 – 2-3, 2-6, 5-2, 5-4	IDMSDBAN	2-13
DB/DYNAM	1-5, 1-7, 2-14, 2-16 – 2-25, 5-5, 5-7	IDMSDBIO	2-5, 2-10, 2-15, 2-16
ALLOCATE Command	2-18 – 2-20	IDMSDBLU	3-10
BYPASS Command	2-24 – 2-25	IDMSINIT	2-13, 2-15, 2-16
CLOSE Command	2-22 – 2-23	IDMSLDEL	3-10
DEALLOCATE Command	2-20 – 2-21	INSTDPLX	5-2, 5-4 – 5-6
DUPLEX Command	2-24	INSTDYNM	5-2, 5-6
START Command	2-25	installation	2-4 – 2-5, 2-13, 2-15, 3-1, 5-1 - 5-7
SWAP Command	2-23	integration	2-13 – 2-16
DB/INFO	1-7, 5-2, 5-4, 5-6	J	
DB/INFO Command	2-25 – 2-29	journal	
INFO DATABASE	2-26	sequential	1-5, 1-6, 2-10, 2-11, 2-12
INFO DPLXFILE	2-26	disk	1-5, 1-6, 2-5 - 2-8, 2-10 – 2-13, 2-16, 3-1, 3-2, 3-5, 3-10, 3-11
INFO DPLXQUE	2-26	journal archive	1-5, 1-6, 3-10
INFO DPLXTABL	2-26	JOURNAL/DUPLEX	1-3, 1-4, 1-5, 1-6, 2-1, 2-6, 2-10, 2-11, 3-10, 5-2, 5-3, 5-4
INFO JOURNAL	2-26	L	
L			
local mode	1-5, 2-10, 2-11, 2-13 – 2-15, 3-1, 3-5 – 3-10		

INDEX

M

manual recovery 3-5 – 3-9
Messages
 DB/DYNAM 4-1 – 4-16
 DB/INFO 4-18 – 4-19
 TDS 4-19 – 4-21
Mirroring 1-1, 1-3 – 1-4, 2-6, 2-10,
2-11, 2-12

N

normal operation 2-4, 2-13, 3-1, 3-2, 3-5, 3-7,
3-9
nonstop operation *See XPO 24X7
Enhancement*

O

Online Component 1-4, 2-1, 2-2, 2-5 - 2-8,
2-10, 2-12, 2-13, 2-14

P

Performance 1-1, 1-3, 1-4, 2-2, 2-4, 2-9,
2-10, 2-12, 2-13, 2-15
procedure
 installation 2-5, 2-15, 5-1 – 5-6
 recovery 3-1 – 3-12

R

recovery operation 1-6, 3-1 – 3-12
 Central Version 3-1 – 3-5
 local mode 3-5 – 3-9
 utility 3-9 – 3-12
restartable 3-9, 3-10

T

TDSUTIL1 2-12
Technetron Duplex Subsystem
 see XPO - TDS
Technetron Dynamic Facility *see XPO - TDF*

U

user abend
 TDF 4-19 – 4-20
 TDS 4-20 – 4-21
user exit 1-5, 2-16, 2-17

X

XPO
 24x7 Enhancement 2-1 – 2-5
 TDF 1-7
 TDS 1-4 – 1-6