TRANSFER™ Delivery System
Programming Guide

Data Management Library

82525
Effective with the B00/E08 software release, Tandem introduced a more formal nomenclature for its software and systems.

The term "NonStop 1+™ system" refers to the combination of NonStop 1+ processors with all software that runs on them.

The term "NonStop™ systems" refers to the combination of NonStop II™ processors, NonStop TXP™ processors, or a mixture of the two, with all software that runs on them.

Some software manuals pertain to the NonStop 1+ system only, others pertain to the NonStop systems only, and still others pertain both to the NonStop 1+ system and to the NonStop systems.

The cover and title page of each manual clearly indicate the system (or systems) to which the contents of the manual pertain.
TRANSFER™ Delivery System
Programming Guide

Abstract
This manual provides reference material and guidelines for application programmers who are writing programs that use the TRANSFER Delivery System.

Product Version
TRANSFER B00

Operating System Version
GUARDIAN A05 or later (NonStop Systems)
GUARDIAN E06 or later (NonStop 1+ System)

Part No. 82525 A00

March 1985

Tandem Computers Incorporated
19333 Valico Parkway
Cupertino, CA 95014–2599
DOCUMENT HISTORY

<table>
<thead>
<tr>
<th>Edition</th>
<th>Part Number</th>
<th>Operating System</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Edition</td>
<td>82325 A00</td>
<td>GUARDIAN A04/E05</td>
<td>April 1983</td>
</tr>
<tr>
<td>Second Edition</td>
<td>82325 B00</td>
<td>GUARDIAN A04/E05</td>
<td>December 1983</td>
</tr>
<tr>
<td>Third Edition</td>
<td>82525 A00</td>
<td>GUARDIAN A05/E06</td>
<td>March 1985</td>
</tr>
</tbody>
</table>

New editions incorporate all updates issued since the previous edition. Update packages, which are issued between editions, contain additional and replacement pages that you should merge into the most recent edition of the manual.

Copyright © 1983 by Tandem Computers Incorporated.
Printed in U.S.A.

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

The following are trademarks or servicemarks of Tandem Computers Incorporated:

- AXCESS
- BINDER
- CROSSREF
- DDL
- DYNABUS
- DYNAMITE
- EDIT
- ENABLE
- ENCOMPASS
- ENCORE
- ENFORM
- ENSCRIBE
- ENTRY
- ENTRY520
- ENVOY
- EXCHANGE
- EXPAND
- FOX
- GUARDIAN
- INSPECT
- NonStop
- NonStop 1+
- NonStop II
- NonStop TXP
- PATHWAY
- Nonstop 1+ Nonstop II Nonstop TXP PATHWAY
- PCFORMAT
- PERUSE
- SNAX
- Tandem
- TAL
- TGAL
- THL
- TIL
- TMF
- TRANSFER
- T-TEXT
- XRAY
- XREF
- XRAY
- XREF

INFOSAT is a trademark in which both Tandem and American Satellite have rights.

HYPERchannel is a trademark of Network Systems Corporation.

IBM is a registered trademark of International Business Machines Corporation.
CONTENTS

PREFACE ................................................................. ix

SECTION 1. A PROGRAMMER'S VIEW OF TRANSFER .................... 1-1
TRANSFER Applications ................................................ 1-3
   The Advantages .................................................... 1-5
   The Limitations ................................................... 1-6
TRANSFER The TRANSFER Environment ................................ 1-7
   Types of Application Processes ................................ 1-8
TRANSFER Objects .................................................... 1-9
   Correspondents .................................................... 1-9
   Packages ............................................................ 1-10
   Depots ............................................................... 1-13
      Distribution Lists .............................................. 1-14
   Profiles ............................................................ 1-15
   Folders ............................................................. 1-15

SECTION 2. TRANSFER APPLICATION COMPONENTS ..................... 2-1
Clients ................................................................. 2-3
Agents ................................................................. 2-4
Specialized Servers .................................................. 2-5
TRANSFER Interactive Server (TISERV) ................................ 2-5
TRANSFER Asynchronous Processes ................................ 2-6
   Scheduler ........................................................... 2-7
   TRANSFER Asynchronous Requesters (TAREQs) .................. 2-7
Interprocess Communication .......................................... 2-8
TRANSFER Sessions ................................................... 2-9
TRANSFER Object Management ....................................... 2-11
   Identifying Packages and Items .................................. 2-11
   Resolving Object Names ......................................... 2-12
      Correspondent Registration .................................. 2-12
      Session Establishment ....................................... 2-13
      Requests on Behalf of a Correspondent .................... 2-13
   Package Delivery ................................................ 2-14
   Moving Packages Between Depots ................................ 2-14
   Inter-object Relationships ..................................... 2-16
   Depot Security ................................................... 2-17
## Contents

### SECTION 3. TRANSFER NAMING CONVENTIONS
- Correspondent Names .................................. 3-1
- Folder Names .......................................... 3-2
- Distribution List Names ................................ 3-7
- Wildcard Names and Patterns .......................... 3-9
- Node Designation ....................................... 3-10
- Name Length Restriction ................................ 3-10
- TRANSFER Name Directory .............................. 3-11
- Name Searching ......................................... 3-12

### SECTION 4. INTERFACING WITH TRANSFER
- Starting a TISERV Process .............................. 4-1
- TISERV Interface ........................................ 4-5
- Request and Reply Headers ............................ 4-7
- UOW Operations ........................................ 4-12
  - Session Control ....................................... 4-14
  - Item Handling ........................................ 4-14
  - Whole Item Operations .............................. 4-15
  - Record Handling Operations ....................... 4-16
  - Item Tree Operations ................................ 4-17
- Package Handling ....................................... 4-18
  - Submittal Preparation .............................. 4-18
  - Recipient List Definition .......................... 4-20
  - Package Delivery .................................... 4-21
  - Package Receiving ................................... 4-22
- Folder Manipulation .................................... 4-22
- TRANSFER Configuration Inquiry ..................... 4-22
- Administration ......................................... 4-23
  - Session Management .................................. 4-23
  - Depot Management .................................... 4-23
  - Distribution List Management ...................... 4-24
  - Name Management .................................... 4-24
- UOW Summary Table ................................... 4-25
- TAREQ Interface ........................................ 4-33
  - Agent Selection ....................................... 4-33
  - Agent Notification Message .......................... 4-34
  - TAREQ Event Packages ................................ 4-37
- Delivery Errors ......................................... 4-41
- Precautions about Using Agents ....................... 4-41

### SECTION 5. UOW DESCRIPTIONS
- UOW Header and Return Code .......................... 5-1
- Software Provided with the TRANSFER Delivery System 5-3
- UOW Definitions ........................................ 5-4
  - ACK-RECEIPT ......................................... 5-5
  - ADD-ITEM-REC ........................................ 5-7
  - ADD-MEMBER .......................................... 5-10
  - ADD-RECIPI ........................................... 5-13
  - ALTER-AGENT-SELECT ............................... 5-16
  - ALTER-FOLDER-ORDER ................................ 5-21
  - ALTER-ITEM-DESCR ................................... 5-26
  - ALTER-PROFILE-ELEM .................................. 5-33
  - ATTACH-COMPNT-A01 .................................. 5-37
  - CANCEL-PKG ........................................... 5-40
COPY-ITEM .................................................. 5-42
CREATE-DEPOT ............................................... 5-44
CREATE-DLIST ............................................... 5-46
CREATE-FOLDER ............................................. 5-48
CREATE-FOLDER-B00 ......................................... 5-50
CREATE-ITEM ............................................... 5-54
DELETE-DEPOT ............................................... 5-56
DELETE-DLIST ............................................... 5-58
DELETE-FOLDER ............................................. 5-60
DELETE-ITEM-REC ........................................... 5-63
DELETE-MEMBER ............................................. 5-65
DELETE-PROFILE-REC ....................................... 5-68
DELETE-RECIPI ............................................. 5-70
DETACH-COMPNT ............................................. 5-72
END-SESSION ................................................ 5-74
GET-AGENT-SELECT ......................................... 5-75
GET-CONFIG-NAME ........................................... 5-79
GET-FOLDER-ORDER .......................................... 5-82
GET-ITEM-COMPNT-A01 ....................................... 5-86
GET-ITEM-DESCR ............................................ 5-89
GET-ITEM-REC ............................................... 5-94
GET-NEXT-SESSION .......................................... 5-99
GET-PROFILE-ELEM .......................................... 5-101
GET-RECIPI ................................................ 5-144
NOOP .......................................................... 5-151
READ-NEXT-MEMBER ......................................... 5-153
READ-NEXT-NAME ............................................ 5-157
READ-PROFILE-REC .......................................... 5-162
READ-PROF-REC-A02 ........................................ 5-165
SAVE-ITEM ................................................... 5-170
SAVE-ITEM-B00 .............................................. 5-173
SAVE-ITEM-BY-KEY .......................................... 5-177
SCAN-FOLDER ................................................ 5-182
SCAN-FOLDER-B00 ............................................ 5-185
SCAN-FOLDER-BY-KEY ........................................ 5-188
START-SESSION .............................................. 5-198
SUBMIT-PKG .................................................. 5-202
UNSAVE-ITEM ................................................ 5-204
WHERE-SAVED ................................................ 5-207
WRITE-PROFILE-REC ......................................... 5-210

SECTION 6. DEVELOPING TRANSFER APPLICATIONS ........... 6-1
Application Development Steps ............................ 6-1
Designing and Writing a Client ............................ 6-15
Grouping UOWs in IPCs ..................................... 6-16
Communicating with TISERV ................................ 6-18
Using Correspondent Names for Distribution Lists .... 6-19
TMF Guidelines .............................................. 6-19
Network Guidelines ........................................ 6-20
SCREEN COBOL Coding Guidelines ........................ 6-20
Debugging Techniques ...................................... 6-21
Designing and Writing an Agent ............................ 6-22
Assigning Names to Agents ................................ 6-25
## Contents

5-6 Storage Layout for GET-PROFILE-ELEM and ALTER-PROFILE-ELEM ....................................... 5-107  
5-7 SCAN-FOLDER-BY-KEY UOW Example .......................................................... 5-195  
6-1 Complete Transactions Involving Extended Processing .................................................. 6-4  
6-2 Application Input/Output Screen ............................................................................... 6-6  
6-3 Avoiding a Nesting Problem ....................................................................................... 6-9  
6-4 Ways in Which Errors Are Reported ......................................................................... 6-12  
6-5 An Agent Monitors Package Delivery ..................................................................... 6-14  
6-6 Component Cycle ...................................................................................................... 6-37  
7-1 Client's Logon Screen ............................................................................................... 7-2  
7-2 Client's Function Menu Screen .................................................................................. 7-3  
7-3 Client's Mail Posting Screen ..................................................................................... 7-4  
7-4 Client's Mail Display Screen ...................................................................................... 7-6  
7-5 SCREEN COBOL Code for Client .................................................................................. 7-9  
8-1 SCREEN COBOL Code for Agent .............................................................................. 8-3  
9-1 Interaction Between TRANSFER Clients .................................................................. 9-2  
9-2 Queue Management Interaction .............................................................................. 9-5  
B-1 Standard Send and Reply Packages ....................................................................... B-9  
C-1 Application with Processes Outside PATHWAY ...................................................... C-2

## TABLES

4-1 TISERV Parameters at Startup Time ................................................................. 4-2  
4-2 TISERV UOW Summary ....................................................................................... 4-25  
4-3 TAREQ Events ........................................................................................................ 4-38  
5-1 Depot Profile File Elements and Access Privileges .............................................. 5-111  
5-2 DDL Data Definitions for Profile Data Elements with GET-PROFILE-ELEM Return Codes ................................................................. 5-116  
5-3 ALTER-PROFILE-ELEM Return Codes .............................................................. 5-139  
9-1 Queue Management UOW Summary ................................................................... 9-16  
9-2 Queue Manager Server Parameters at Startup Time ........................................... 9-32  
9-3 Queue Manager ASSIGN Command Parameters at Startup Time ........................... 9-36  
9-4 Queue Management Initialization Error Messages .............................................. 9-38  
A-1 GUARDIAN File Codes .......................................................................................... A-5  
A-2 Error Codes ........................................................................................................... A-6  
A-3 Alphabetic Listing of Error Codes ....................................................................... A-59  
B-1 Effect of WORKSPACE-ITEM and COMPONENT-ITEM Settings ..................... B-4  
D-1 TAREQ Events ........................................................................................................ D-1
PREFACE

This manual is a guide to writing application programs that use the TRANSFER delivery system for the Tandem NonStop 1+ System and NonStop Systems. Such programs cooperate with TRANSFER in moving information throughout a single computer system or a network of distributed systems. The programs typically run under the PATHWAY transaction processing system and use the services of the Transaction Monitoring Facility (TMF) to ensure transaction consistency.

This manual is intended for application programmers. The manual presents the concepts, guidelines, and detailed reference information needed to support application program interaction with TRANSFER; and also includes a description of the tools that provide access to the features and capabilities of TRANSFER.

The following manuals contain more detailed information about the Tandem NonStop 1+ System and NonStop Systems and the software products used with TRANSFER:

- **Introduction to Tandem Computer Systems** provides an overview of the NonStop 1+ System and NonStop Systems hardware and software.

- **PATHWAY SCREEN COBOL Reference Manual** describes the SCREEN COBOL programming language that is used in the development and control of online transaction processing applications.

The following manuals contain detailed information for programs that are written in a language other than SCREEN COBOL or that interface with programs written in other languages:

• GUARDIAN Operating System Programming Manual, Volumes 1 and 2 provide information about interfacing programs with the GUARDIAN operating system and accessing the hardware and software resources of the system.

This manual is a member of the TRANSFER Delivery System Manual Library. Other manuals in this library include:

• Introduction to TRANSFER Delivery System is an overview of TRANSFER.

• TRANSFER Delivery System Management and Administration Guide is a guide to managing and administering the TRANSFER Delivery System.

• TRANSFER/MAIL Users Guide is a guide to using the TRANSFER/MAIL application supplied by Tandem.
SECTION 1

A PROGRAMMER'S VIEW OF TRANSFER

The TRANSFER delivery system is a high-level software product that reliably supports communications between people, input/output devices, and processes. The product is especially useful in situations where system resources are widely distributed or intermittently available, or both.

In addition to providing a standard electronic mail application, TRANSFER/Mail (T/MAIL), TRANSFER supports a wide spectrum of user-written applications. These are typically business applications that link interactive environments and involve such features as:

- transmittal of data to multiple destinations, such as routing survey questionnaires and newsletters, or updating distributed, replicated data bases
- communication among applications with different missions, or among components of a distributed application, such as a manufacturing system transmitting data to a purchasing system
- scheduling of transactions for future occurrence, or transactions of long duration, such as generating a report.

Figure 1-1 illustrates information delivery with TRANSFER.

TRANSFER appears as a group of processes that your programs (also running as processes) can call upon to build, maintain, and route collections of information called packages throughout your computer system or network. TRANSFER maintains a data base that describes TRANSFER users and provides them with holding areas for packages and parts of packages. All interactions between your application processes and TRANSFER take place through interprocess messages.
TRANSFER is independent of package content. Any application program can request the services of TRANSFER to create, build, request transmittal of (post), store, and access packages. Your programs make these requests via the TRANSFER programmatic interface.

An application can have interactive communication with users, or can have one process interact with another process with no human intervention. An application can contain programs written in various languages, all using TRANSFER in one way or another; for example, a PATHWAY requester written in the SCREEN COBOL language could use TRANSFER to transmit data to a statistical application written in FORTRAN.
TRANSFER APPLICATIONS

TRANSFER is suitable for many different kinds of applications that involve information transmittal. TRANSFER is particularly valuable for dealing with two basic characteristics commonly found in business transactions: end-to-end communication and extended transaction processing.

- In end-to-end communication, a correspondent, which can be a person, input/output device, or process, sends a package to one or more other correspondents. Although the sending and receiving correspondents might all exist at the same network node, the interaction often takes place between nodes and involves distributed resources. TRANSFER can send a package to multiple destinations anywhere in the network.

As shown in Figure 1-2, a simple instance of end-to-end communication occurs when an operator at a terminal composes a memo and then sends it to a person at another terminal. A more sophisticated example might involve a manufacturing application running at one site and interacting with a purchasing application at another. If you were developing this type of application without benefit of TRANSFER, you would have to write software to take care of several contingencies—for example, what to do if a package is sent to a node that is unavailable, or to a correspondent who is either unavailable or not ready to receive the package. That software would have to monitor network availability, queue packages, and manage timing considerations; but with TRANSFER, these functions are managed for you by software supplied by Tandem.

- In extended transaction processing, a sender initiates one transaction and then proceeds immediately to a new one without waiting for work on the original transaction to be completed. Figure 1-3 illustrates extended transaction processing. This is frequently called nowait transaction processing because the application continues with another transaction without waiting for the first one to be completed.
For an example of extended transaction processing, consider an application that:

a. allows an operator to request the system to build an inventory report from various items in a data base and to print the report on a local printer that will not be free until 4:30 p.m. that afternoon

b. then, permits the operator to send memos to interactive users, announcing the pending availability of the report, while the system is gathering the data for it.

This kind of transaction processing distributes work over both time and geography. It promotes great efficiency, allowing operators and applications to move on to other work while TRANSFER completes the work they originally started.
A Programmer's View of TRANSFER

Figure 1-3. Extended (Nowait) Transaction Processing

The Advantages

TRANSFER relieves application programs of several tasks that they would otherwise have to perform. Among these are:

- delivering packages composed of multiple components
- monitoring the computer network for availability
- managing the routing of packages to multiple recipients throughout the network
- ensuring reliable package delivery, even in the event of failures, while preventing duplicate deliveries
- providing temporary storage of items and packages.
A Programmer's View of TRANSFER

An application can specify the earliest and latest times at which a package can be delivered. If the package cannot be delivered within that time window, TRANSFER returns it to the sender. An application also can specify priorities for packages so that if many packages are posted within a short time, those with the highest priorities are processed first.

Recipients of packages need not be available to receive them. If a network node is unavailable when TRANSFER attempts to deliver a package, delivery takes place as soon as the node does become available. If the recipients are unavailable, TRANSFER stores packages for those recipients, who can collect them at their own convenience.

When an application requests TRANSFER to deliver a package, that package will reach its destination exactly once. In the event of a hardware failure, TRANSFER ensures that a package is neither lost nor delivered twice to any recipient. An application also can define certification criteria by specifying, for example, that a sender is always notified in some way when all recipients have received packages that were posted.

The Limitations

Although TRANSFER is a powerful application tool, it does have some limitations:

- TRANSFER cannot ensure that any package will be delivered at a specific time, and cannot predict how much time a package will spend in transit.

- TRANSFER does not attempt to deliver packages in the same order that they are sent; your application program, however, could arrange to process packages, upon receipt, in the order in which they were transmitted.

- Although TRANSFER allows a package to contain any kind of data, TRANSFER does not interpret or process the data in the packages. That is the responsibility of your application program.

As an application programmer, you must determine:

- what constitutes a transaction and what signifies its completion

- what kinds of information a package contains, and in what arrangement

- who receives the packages
• how recipients process and respond to packages

• how correspondents are related to one another, and how packages are interrelated.

An important relationship to consider is that between two common application requirements:

the need for packages to be delivered within a narrow timeframe

the need for a certain percentage of packages to be delivered successfully.

The narrower the timeframe, the greater the chance that external factors might prevent package delivery. If an application requires both a narrow timeframe and a high percentage of delivery, you should carefully consider the overall capabilities of TRANSFER when deciding whether to use this product.

A final limitation is the fact that an application cannot control traffic on network lines. Although an application can defer delivery of packages, it cannot defer transport of those posted for remote nodes. The only way to defer transport of a package is to either avoid posting it or shut down the network. For example, if you want to defer transport until after normal working hours, you could have an operator post a triggering package for nighttime delivery, and then have an application process post the real package when it receives the trigger.

THE TRANSFER ENVIRONMENT

An application can run under the PATHWAY transaction processing system, taking advantage of the requester/server model on which all PATHWAY applications are based. In the PATHWAY environment, the application programs are requesters written in the SCREEN COBOL programming language and are used to control terminal input/output. In general, requesters for a TRANSFER application fit into the operating environment in the same way as requesters for other PATHWAY applications.

Outside the PATHWAY environment, application programs can be modules coded in COBOL, FORTRAN, or the Transaction Application Language (TAL).

TRANSFER uses the Transaction Monitoring Facility (TMF) to ensure the consistency of any files that are changed during the delivery process. If a failure occurs while a package is in transit, the delivery operation is backed out and started over again. In the event of a system failure, TMF allows you to reconstruct the files.
A Programmer's View of TRANSFER

Types of Application Processes

A TRANSFER application consists of a set of processes that communicate with each other and with TRANSFER; they work together to perform a common task. These processes might run on different CPUs or even at different nodes in a network. The entire application includes:

- TRANSFER processes--processes that are furnished as part of TRANSFER, and processes that are provided by PATHWAY and operate in conjunction with TRANSFER
- User processes--processes for which you, the application programmer, write the code.

Basically, you are concerned with writing programs for two general kinds of user processes: clients and agents. Each type is defined by the task it performs.

- Clients are requester programs that provide the interface between correspondents and TRANSFER. Clients allow the user to build, alter, and post packages for delivery, and receive and respond to incoming packages. They also allow the user to request various administrative functions. In terms of the PATHWAY requester/server model, clients are those requesters that make requests of servers furnished as part of TRANSFER. Clients can also communicate with any number of user-defined servers.

  Most clients are written as SCREEN COBOL programs that execute in PATHWAY Terminal Control Processes (TCPs). Others are programs written in COBOL, FORTRAN, or TAL; these processes, however, cannot take advantage of PATHWAY features such as load-balancing operations, ease of coding interprocess communications and TMF requests, and resource management capabilities.

- Agents allow applications to participate in the delivery process. Agents are programs that are automatically invoked to handle packages received at a depot. A depot is that portion of the TRANSFER data base associated with a particular correspondent. When a package arrives at the depot, the agent is notified by TRANSFER. In response to this notification, the agent usually takes some kind of action, such as filing a package for later retrieval, initiating a transaction, or replying to the sender. Basically, the agent eliminates the necessity for polling to see if packages have arrived. The agent provides a facility for automatically processing incoming packages without the need for human intervention.
Agents can be written as PATHWAY requesters in SCREEN COBOL, or as server processes in COBOL, FORTRAN, or TAL. In the role of a requester of TRANSFER services, an agent can in fact be regarded as a kind of specialized client.

Clients and agents sometimes perform many of the same functions, and they are not always totally distinct. They are treated separately in this manual because they usually involve different design and coding considerations. The most fundamental distinction between these two types of processes is:

- Clients take their control information either from user processes or from other processes that are not part of TRANSFER. They direct most of their output to TRANSFER.

- Agents take their control information from TRANSFER and the received packages that trigger them. They direct output to processes or devices external to TRANSFER, or back into TRANSFER itself.

TRANSFER Objects

A TRANSFER application involves the following objects:

- correspondents that send and receive packages
- the packages themselves
- depots where packages are delivered
- distribution lists that allow packages to be sent to multiple destinations
- profiles that describe the correspondent
- folders where packages are stored.

CORRESPONDENTS. A correspondent can be a person, a process, or a device as illustrated in Figure 1-4. Each correspondent is assigned a unique name that explicitly identifies both the node where the correspondent receives packages and the specific identity of the correspondent. TRANSFER keeps track of all correspondent names in its own data base.
PACKAGES. A package is a collection of information that can be sent from one correspondent to another correspondent. A person-to-person package could be interoffice mail. A process-to-process package could be data and transaction codes needed to update a database.

Packages are made up of discrete collections of data known as items. Each item always includes an item descriptor, which is composed of one or more records that describe the attributes and composition of the item. In addition, the item can include one or more data records. Examples of items are shown in Figure 1-5.
An item usually contains only one kind of data whether that data is text, ASCII data, digital facsimile data, or some other type defined by your application. Each item is identified by its own unique item ID. Items can range in size from a few to many thousands of bytes and can also contain other (component) items.

Any process representing the correspondent who created an item can add, delete, examine, or change the contents of the item until it becomes part of a submitted package. For example, a process can retrieve a data record, insert records, and change their order. Similarly, a process can modify components in any list of component items that is part of the item. A process cannot, however, delete or change an item if the item is part of a package that is already posted.

In order for TRANSFER to deliver items from one correspondent to another, the items must be assembled as packages as shown in Figure 1-6.
A Programmer's View of TRANSFER

![Diagram of package structure]

Figure 1-6. A Package

Each package includes a package header item, perhaps one or more records or component items, and a list of recipients. The package header item is like the label on an actual mail package; it indicates who is sending the package, and to whom, when the package was posted, and what its delivery priority is. The package header might also indicate the timeframe in which the package should either be delivered or expire; it also might include a components list that contains the IDs of all items in the package. If an item in a package contains other items, those second-level items are not listed in the package header.

An application builds a package by creating the package header item and making separate requests to add recipients, component items, and delivery parameters. The package itself might be composed of different kinds of items in any arrangement. For example, a package could contain a formatted ASCII item and a digital facsimile data item, along with other package header items. If the components list of a package includes another package header item, the result is a package nested in another package, as shown in Figure 1-7. A package that contains another package is just a specific instance of the general rule that an item can contain other items.
DEPOTS. A depot is that portion of a TRANSFER data base associated with a particular correspondent. The depot is established when an application registers the correspondent with TRANSFER. Every correspondent has precisely one depot and that depot has a network-unique identity, which an application references implicitly through the correspondent name. A depot principally contains distribution lists, profiles, and folders, as shown in Figure 1-8.
DISTRIBUTION LISTS. A distribution list simplifies the delivery of a package to multiple correspondents. The list can contain the names of both individual correspondents as well as the names of other distribution lists. This allows an application to send packages to multiple destinations, while referencing those destinations by only one common name. As long as a correspondent knows a particular distribution list name, that correspondent can remain totally unaware of the specific names on the list.

Before a package can be delivered to the correspondents on a distribution list, that list must be expanded. To accomplish this expansion, TRANSFER replaces the referenced distribution list name with the actual names of the correspondents on the list. These correspondents are known as the members of the distribution list.
PROFILES. A profile is a set of one or more records that describe the correspondent who owns the depot. These records reside in a profile file, which contains the profiles for all correspondents registered at the node. These records are managed by TRANSFER.

A correspondent profile contains identifying data, such as the password of the correspondent and operational parameters applicable to the correspondent. The profile also indicates privileges of the correspondent and the default values for package delivery parameters, such as delivery priority and expiration time.

An agent profile specifies the agent or agents that should be invoked when a package arrives at the depot. The profile indicates the name of the agent; selection criteria for invoking the agent; and other information, such as whether the agent is a SCREEN COBOL requester program or a server.

TRANSFER also manages other profile records for each depot. These include profile records for T/MAIL and any other applications requiring storage of data applicable to the depot.

Through requests to TRANSFER, an application can examine the profile records for the correspondent that it is representing, and can alter certain fields within these records. An application that represents someone with system administrator capability can examine and change certain fields in the profile records of other correspondents at the node. A system administrator is a person who is responsible for managing the objects that TRANSFER uses in the delivery process.

FOLDERS. A folder is an area where related items and packages are stored. As an example, a correspondent might keep separate folders for packages received from different sources, separate folders for text and facsimile data, or a separate folder for packages that are to be sent in response to earlier packages. The same item or package can belong to more than one folder.

An application process can create and delete folders, add items and packages to them in a predefined ordering sequence, and examine their contents. In addition to folders created and maintained by application processes, TRANSFER maintains the following special folders:

- INBOX Folder. When a package arrives at a depot, TRANSFER saves it in a folder named INBOX. Packages remain in this folder until they are explicitly removed or until they expire, whichever event occurs first. Once TRANSFER places a package in a recipient's INBOX folder and invokes the appropriate agents, delivery of the package to that recipient is considered complete. Clients can explicitly remove packages
A Programmer's View of TRANSFER

from the INBOX. If a client never acknowledges receipt of a package, TRANSFER removes it from the INBOX and returns it to the sender when the package expiration time occurs. An application can retrieve any items from this special folder.

• WASTEBASKET Folder. A process that intends to delete an item at the end of a session can save the item in a folder named WASTEBASKET. A session is defined as the period of time during which a correspondent can submit requests to TRANSFER. When the process ends the session, the item is automatically purged from this folder. An application might use WASTEBASKET so that an operator who mistakenly deletes an item during a session can retrieve the item before the end of that session. An application can retrieve any items from this special folder.

• OUTBOX folder. This folder is not currently used by TRANSFER, but is reserved for future use. At present, it is illegal for any process to save a package in the OUTBOX.
SECTION 2
TRANSFER APPLICATION COMPONENTS

The primary processes in a TRANSFER application are:

- clients
- agents
- specialized servers required by clients and agents
- TRANSFER interactive server
- TRANSFER asynchronous processes
- processes that communicate and operate in conjunction with TRANSFER

When these processes run under the PATHWAY transaction processing system, they take advantage of the requester/server model on which all PATHWAY applications are based. These processes and the way in which they interact within the framework of PATHWAY are illustrated in Figure 2-1.
TRANSFER Application Components

Shading indicates processes supplied by TRANSFER.
Dashed lines indicate processes that are part of the TRANSFER system.

Figure 2-1. Processes in a TRANSFER Application
CLIENTS

Clients are requester programs that provide the interface between correspondents and TRANSFER. They allow correspondents to communicate and interact with TRANSFER. A client, for example, enables a correspondent to create new items, assemble them into packages, post these packages, retrieve incoming packages, and request various administrative services. For every TRANSFER application, you must code one or more clients.

A client typically performs the following functions:

- establishes and terminates communication with TRANSFER
- manages the terminal interface on the basis of the screens you design and the code you write to accept and display data on those screens
- composes items and packages at the depot
- assigns delivery parameters, such as package priority and delivery timeframe, and posts packages for delivery
- takes certain actions based upon receipt of packages, such as presenting a message to a user at a terminal or starting another transaction
- detects and reacts to certain error conditions.

A client can represent any number of correspondents; TRANSFER recognizes a correspondent as a name with an associated depot in the TRANSFER data base.

Examples of clients in TRANSFER applications are:

- SCREEN COBOL programs operating under a PATHWAY terminal control process (TCP), where the SCREEN COBOL programs
  - represent each operator as a correspondent
  - represent each terminal as a correspondent
  - represent all terminals and operators as a single correspondent
  - represent other servers or devices handled by those servers as correspondents.

- any GUARDIAN process that makes requests of TRANSFER servers on behalf of correspondents.
TRANSFER Application Components

Most clients are written as SCREEN COBOL programs that execute within PATHWAY TCPs. Clients can be written as FORTRAN, COBOL, or TAL programs, but they cannot take advantage of the resource management capabilities of PATHWAY.

AGENTS

Agents are SCREEN COBOL requester programs or PATHWAY server classes that are automatically invoked to handle packages received at a depot. Agents are invoked by TRANSFER asynchronous requesters (TAREQs), which are responsible for actual package delivery. An arriving package might have been sent by another correspondent. An arriving package also might have been sent by TRANSFER itself; for example, an incoming package from TRANSFER could be certifying delivery of an earlier package, or could be telling the correspondent that a package delivery time expired before a remote node became available.

The capability of defining agents is an important feature of TRANSFER. Without an agent, a correspondent must check to determine whether packages have arrived at the depot; with an agent, the package arrival can automatically trigger some action. For example, an agent can react to a package delivery by:

- informing the correspondent of the delivery
- saving the package automatically in a folder at the depot
- initiating a transaction, perhaps in response to a triggering package
- creating a new package from the package received, adding items to that package, and forwarding it to another correspondent
- arranging for packages to arrive in a certain order
- sorting packages according to type before presenting them to the receiving correspondent
- filing packages from different applications separately, with the client then allowing the correspondent to retrieve the packages selectively
- replying automatically to the sender
- displaying the package on a device
- deleting packages that are not meaningful to the receiver.

An agent is selected for execution by criteria supplied in the agent profile associated with the depot. The criteria can be
TRANSFER Application Components

established either by a client or by an operator using the ADMIN application. ADMIN is a TRANSFER application, supplied by Tandem, that provides a user interface to the administrative functions of TRANSFER.

A single agent can be associated with several depots. A single depot can have many agents, each with different selection criteria, so different types of packages and applications can be handled by different agents.

An agent can be a requester or a PATHWAY server. In general:

- An agent that makes requests of TRANSFER server classes to retrieve, save, or post packages should be a requester written in SCREEN COBOL.

- An agent that accesses an input/output device, or that performs string handling, complex computations, or tasks of long duration should be a server written in COBOL, FORTRAN, or TAL.

- An agent that handles both kinds of work should be written in SCREEN COBOL and should direct its requests to a special application-defined server class.

Agents are invoked by TAREQs according to information in the incoming package and in the recipient's profile. SCREEN COBOL agents run in the same TCPS as the TAREQs that invoke them. Communication between TAREQ and agent is program unit to program unit, not TCP to TCP.

SPECIALIZED SERVERS

Your clients and agents might require the support of specialized servers. You can write these servers in COBOL, FORTRAN, or TAL. You can also write servers in the Massachusetts General Hospital Utility Multi-Programming System (MUMPS) provided the servers require no input/output audited by the Transaction Monitoring Facility (TMF).

TRANSFER INTERACTIVE SERVER (TISERV)

Any TRANSFER application always has a TRANSFER interactive server class defined at each node. This server class is called interactive because its primary job is to receive and reply to requests from your clients. The server is usually named TISERV.
TRANSFER Application Components

TISERV furnishes the interface between TRANSFER and your clients. This server performs the following:

- handles all the work involved in accessing the TRANSFER data base on behalf of your clients
- contacts the TRANSFER asynchronous processes when a client has posted a package for delivery.

Any application can include servers in addition to those supplied by TRANSFER, and can include its own data base as well.

TISERV receives requests from requesters, performs the services requested, and replies. These requesters include the clients and agents you write for your application, applications provided by Tandem, and the TRANSFER asynchronous processes provided by TRANSFER.

TISERV handles requests to:

- create new packages, items, and folders
- add, delete, and retrieve information from packages, items, and folders
- post packages
- change profiles and distribution lists
- add or delete correspondents
- perform administrative tasks.

TRANSFER servers are context free; thus, requests on behalf of a particular correspondent need not all go to the same server process. You indicate in the code for your requester what server class should receive the request, and PATHWAY selects a specific server process from that class. A server locks every record it updates during a transaction; still, it is possible for several server processes to perform parts of the same transaction. Every process that does work for one particular transaction uses the same TMF transaction identifier (TRANSID); thus, a lock set by one server does not prevent access to the record by another server.

TRANSFER ASYNCHRONOUS PROCESSES

TRANSFER asynchronous processes handle package delivery. They schedule packages for delivery, locate their recipients, and accomplish their transmittal. They ensure that a package is
TRANSFER Application Components

delivered within the timeframe specified by the sender; if a network node is unavailable, they ensure packages destined for the node are transported as soon as the node becomes available. They also cancel packages and arrange for their expiration.

TRANSFER provides two kinds of asynchronous processes: a scheduler and the TRANSFER asynchronous requesters (TAREQs).

User-written client requesters do not communicate directly with the asynchronous processes. To post a package, a client sends its request to TISERV; TISERV, in turn, contacts the scheduler.

The only case in which asynchronous processes communicate directly with user-written software is when TAREQs invoke agents.

Scheduler

Every node has a scheduler that runs as a fault-tolerant process pair. The scheduler, of which only one copy exists, maintains queues on disc to keep track of the following:

- packages that are ready for delivery, expiration, or cancellation
- packages that cannot be acted upon because their timestamps call for deferred action
- packages waiting for delivery, expiration, or cancellation at unavailable nodes.

Packages move to the ready queue when all conditions for delivery are met. When a network node is available, all packages that have recipients at that node are placed in the ready queue. If some recipients are at available nodes and other recipients are at unavailable nodes, only transport to the unavailable nodes is deferred; the package is considered ready for transport to all available nodes.

When a package is transported to a node, distribution lists local to that node are expanded. TRANSFER replaces the distribution list name with the actual names of the correspondents; this action might also result in transport of the package to other nodes when they are available.

TRANSFER Asynchronous Requesters (TAREQs)

A TAREQ is a collection of SCREEN COBOL programs that handle the actual delivery of a package to a depot. These programs are supplied by Tandem and run within a standard PATHWAY TCP. Every node has at least one TAREQ.
TRANSFER Application Components

TAREQs request work assignments from the scheduler, one assignment at a time for each TAREQ. TAREQs communicate with the scheduler as though the scheduler were a group of conversational terminals; the scheduler, in other words, acts as a terminal simulator. No real terminals are associated with TAREQs.

The TAREQs locate the recipients of packages and deliver the packages to local depots by issuing requests to TISERV. They also arrange for remote deliveries by issuing requests to copies of TISERV at other nodes as shown in Figure 2-2.

![Diagram of package delivery to a remote node]

Figure 2-2. Package Delivery to a Remote Node

INTERPROCESS COMMUNICATION

Clients and agents communicate with TRANSFER processes through interprocess messages.

- A client written in SCREEN COBOL issues SEND statements to TRANSFER server classes. The same SCREEN COBOL program can communicate with other server classes also.
• A client written in COBOL, FORTRAN, or TAL makes requests by opening a TRANSFER server and issuing WRITEREAD requests to that server through the GUARDIAN operating system.

• An agent defined as a PATHWAY server class is invoked by TAREQs issuing SEND statements to the class.

• An agent defined as a requester and written in SCREEN COBOL is invoked by TAREQs issuing SCREEN COBOL CALL statements to the requester.

Agents, like clients, can issue requests to TRANSFER. Agents written in SCREEN COBOL issue requests through SEND statements; agents written in COBOL, FORTRAN, or TAL issue requests with WRITEREAD calls. The format of requests issued by your program is the same, whether the request is issued in a SEND statement or a WRITEREAD call.

To issue requests to TRANSFER, clients and agents must first initiate communication with TRANSFER by establishing a session.

TRANSFER SESSIONS

Before a requester process can access a correspondent depot to build, post, or receive packages, or do other tasks for that correspondent, the requester process must establish a session between the correspondent and TRANSFER.

To establish a session, the requester process sends a request to TISERV, supplying the name and password of the correspondent that the requester represents. TISERV must be running at the node where the correspondent is registered.

A session takes place between a correspondent and TRANSFER. As illustrated in Figure 2-3, the session includes transactions to post a package for delivery, but does not include the routing and delivery of packages to depots. Because of the TRANSFER nowait feature, neither the sender nor any recipient of a package needs to have a session in progress while the package is in transit or when it arrives at a depot. A correspondent terminates the session when services are no longer required for the depot.

When a requester initiates a session, TISERV returns a session ID to the requester. This ID, which is unique throughout the network, must occur in every later request that the requester makes for the correspondent. Eventually the process terminates the session, perhaps because the correspondent logged off, and relinquishes the session ID.
An agent requiring access to a correspondent depot must establish a session just like any other requester, or be granted one by a TAREQ. You can set up a correspondent profile so that an agent is automatically granted a session whenever it is invoked, in which case the agent does not need to furnish the depot password.
Because session IDs are unique at the node and TRANSFER servers are context free, the following relationships can exist among requesters, servers, correspondents, and sessions:

- One requester can maintain sessions in progress for several correspondents. Requests for each correspondent carry that correspondent's session ID.

- The same requester can have more than one session for the same correspondent, with different session IDs. The same correspondent, for example, might be a user logged on at two terminals controlled by the same TCP.

- Multiple requesters can maintain sessions in progress for the same correspondent at the same time. Each requester has its own session ID for the correspondent.

- The requests issued during a session need not all go to the same server, nor even to the same server class. The session ID accompanies every TRANSFER request, and any TRANSFER server at the node can interpret it correctly.

TRANSFER OBJECT MANAGEMENT

The various elements that TRANSFER maintains and manages during package creation and transmittal are known as objects. These objects are: correspondents, packages, depots, distribution lists, profiles, and folders.

The following paragraphs describe how TRANSFER manages these objects.

Identifying Packages and Items

Every package or item is assigned an identifier (ID) that is unique throughout the network. This ID always consists of:

- a system number that defines the network node where the package or item is created and ensures uniqueness among IDs on different systems

- a local ID that distinguishes among different objects created at a particular node.

Through this ID, your application can fully identify and access any package or item anywhere in the network without any chance of ambiguity.
TRANSFER Application Components

Resolving Object Names

Just as packages and items must be uniquely identifiable, so must correspondents, distribution lists, and folders. TRANSFER accomplishes this identification through the assignment of names for correspondents, distribution lists, and folders. When a process references one of these objects, TRANSFER must ensure that the process supplied a valid object name even though that name might be an abbreviation for some other name. Name resolution is performed as follows:

1. TRANSFER transforms the given name into one or more fully qualified names, using the name searching method described in Section 3. A fully qualified name is a name that includes the complete name of the correspondent and the network node.

2. TRANSFER validates the fully qualified name by checking the TRANSFER name directory to verify that the name is registered there and that it designates the correct object type (such as correspondent or distribution list). When a new name is registered, the validation operation ensures that the name is unique within the directory before it is added.

Name resolution takes place when:

• a user process registers (adds) a new correspondent to your application

• a user process tries to initiate communication between a correspondent and TRANSFER (called beginning a session)

• a user process supplies one or more names in a request on behalf of a correspondent

• TRANSFER expands a recipient list to deliver a package to one or more recipients.

CORRESPONDENT REGISTRATION. An application must register any new correspondent at the same node as the TISERV process through which the application requests this registration. Registration is requested by a process running on behalf of the system administrator for the node. When an application requests this registration, TRANSFER places the name of the correspondent in the name directory.
SESSION ESTABLISHMENT. To represent a correspondent, a client must first initiate a session for that correspondent; this establishes communication between the correspondent and TRANSFER. The client directs its request for a session to a TISERV server at the node where the correspondent is registered. TRANSFER checks to ensure that the correspondent name is properly registered. TRANSFER then starts the session at that node. The session cannot be passed to servers at any other node.

Agents, like clients, can initiate sessions. Agents can be configured so that when they are invoked, they are automatically granted preestablished sessions. The sessions allow them to make requests to TISERV on behalf of correspondents.

REQUESTS ON BEHALF OF A CORRESPONDENT. When a process supplies a correspondent name as an input parameter to a request, the name is resolved immediately if it is a local name. When a process supplies a correspondent name defined at a remote node, the process must include the name of the node in its request to TISERV; if the remote names are syntactically correct, resolution of the names is postponed if the request specifies deferred resolution.

When a process adds a name to a recipient list or a distribution list, the process can specify whether the name should be resolved immediately or later. The following rules apply:

- If a process requests immediate resolution of a recipient or distribution list member name, that process waits while TRANSFER performs this resolution. If a remote node is required but inaccessible, a resolution error occurs.

- If a process requests deferred resolution of a recipient or distribution list member name, TRANSFER accepts the name without further processing and resolves it later. In most cases, a process only requests deferred resolution when referencing a name defined at a remote node. For remote deferred resolution, TRANSFER still ensures that the name supplied is syntactically correct when it is entered.

If a supplied name is that of a distribution list, the names of individual members are never resolved immediately. Instead, the distribution list is expanded and the member names are resolved during package delivery at each node.

When resolving a name, TRANSFER consults internal directories. TRANSFER first determines whether the name is that of an object belonging to the correspondent represented by the requesting process. If not, TRANSFER then checks to verify that there is a correspondent with that name.
TRANSFER Application Components

Consider a request on behalf of a correspondent named SIMMS-JASON in which the object name MANAGER appears. The object name MANAGER is resolved as follows:

1. Is there an object named MANAGER that belongs to correspondent SIMMS-JASON? MANAGER could be the name of a folder or a distribution list.

2. If not, is there a correspondent named MANAGER at this node?

3. If not, the object named MANAGER does not exist and is an illegal reference.

A programmer might find it difficult to tell at a glance whether an object name is really a correspondent name, folder name, or distribution list name. TRANSFER, however, can readily identify the object type because names are always associated with particular types in the TRANSFER name directory.

PACKAGE DELIVERY. During delivery of a package, TRANSFER processes at the sending node expand local distribution lists and resolve the names of local and remote members. If a package has recipients at remote nodes, a copy of the package is transported to each of those nodes; distribution lists are then expanded and the member names resolved by processes at those nodes as well.

Moving Packages Between Depots

To move items from one depot to another, your application must bind the items into packages. Each package includes a package header item, zero or more component items, zero or more item data records, and a list of recipients.

The package header item specifies:

- the network-unique item ID of the package
- the name of the correspondent sending the package
- timestamps to indicate when the package was posted by the sender, when it should be delivered, what the time zone difference is between the sending node and the receiving node, and when the package should expire
- other package delivery parameters, such as delivery priority and agent selection criteria.
During package delivery, a copy of the package goes to every node where a recipient for that package has a depot.

The term recipient list should not be confused with distribution list.

A recipient list contains the names of correspondents and distribution lists that are to receive the package. The list is an attribute of the package and applies only to the package for which the recipient list is created. The list, which has no name of its own, ceases to exist when the package is deleted.

A distribution list is a predefined list of recipient names; the list provides a simple method for a correspondent to send a package to multiple destinations. A member of a distribution list can be a correspondent or another distribution list. The distribution list belongs to the depot owner and exists independently of any packages.

Initially, the recipient list for a package has one entry for each name supplied by the application. As names are resolved and distribution lists expanded, new names are added to the list and control information is updated.

Once a package is created, only a process that represents the creator of that package can change its contents. Once the package is posted, the application process cannot alter its contents. After the package is posted, local names on the recipient list are validated, local distribution lists are expanded into member names, and duplicate names are eliminated. This process is repeated until all local names are resolved or found to be in error.

For local recipients, TRANSFER delivers the package by placing an entry in the INBOX folder of each recipient and notifying any appropriate agents. For remote recipients, TRANSFER moves a copy of the package to the appropriate remote nodes, and name resolution takes place again at those nodes.

TRANSFER ensures that a package is never moved to a node more than once. Even if a distribution list at a remote node adds new recipients located at the local node, TRANSFER provides delivery to the new local recipients by using the copy of the package that already exists at the local node.
TRANSFER Application Components

INTER-OBJECT RELATIONSHIPS. The use of network-unique IDs for TRANSFER objects provides for important logical relationships among them and efficient use of disc storage space.

- When a user process sends a package to a local correspondent, TRANSFER does not make a copy of the package; instead, TRANSFER makes the INBOX folder of that correspondent point to the package.

- The profiles of all correspondents defined at a particular node are stored in the Profile file.

- The contents of folders are described in the Folder file.

- Records that describe packages and items, called item descriptors, are stored in the Item Descriptor file. The relation between an item (possibly a package) and its component items is specified in the item descriptor.

- The data records for the items are stored in the Item Data file.

- A depot itself is simply a network-unique ID that appears in records of various kinds, distributed over several TRANSFER files.

When an item is saved in multiple folders or even at multiple depots, pointers are also used to avoid duplication of data. This kind of relationship is illustrated in Figure 2-4.

According to this arrangement, several packages can include the same item. Also, several folders can include the item without necessarily including any package of which the item is a part. Finally, the folders that include the item need not all belong to the same depot.

Because each depot and item ID includes the system number of the node where the ID was assigned, it is vital for the system manager at the installation to observe the following:

CAUTION

If you change the system number of a node where TRANSFER has been running, all system IDs stored in the TRANSFER data base must be converted--otherwise, depot and item IDs will point to the wrong depots and items.
TRANSFER Application Components

DEPOT SECURITY. TRANSFER provides different levels of security for different correspondents to prevent unauthorized access to the information maintained by an application:

- Privileges of depot owners. The owner of a depot is, effectively, any person or process who knows the correspondent name and password for logon purposes. Depot owners can:
  - create new folders, distribution lists, items, and packages
  - make additions or changes to any of those objects that they create
  - change many elements of their own profiles.

Depot owners are the only ones who can examine or modify the contents of folders belonging to their depot. Depot owners...
cannot change packages or any component items of packages that are received from other depots or that are posted for delivery. A user process, however, could create a new item that is a copy of the package or item and then change that new item.

- Privileges of system administrators. System administrators can register new correspondents at their node or remove them from the node. Registering a correspondent automatically creates a depot for that correspondent, and removing a correspondent deletes that depot. System administrators can also act on behalf of other correspondents. They can, for example, change the profile of another correspondent (except for selected elements that remain under TRANSFER control); create and delete folders; and create, read, modify, and delete distribution lists at any depot at their node.

- Privileges of other correspondents. Correspondents, including system administrators at other nodes, have very limited access to depots of other correspondents. A process that knows the name of a correspondent can see whether that correspondent has a depot at the node; any process can get a list of all correspondents at the node. Any correspondent can get the names of distribution lists belonging to other correspondents and use the names of the lists; however, a correspondent at one node cannot examine the members of a distribution list at another node.

Correspondent A can use Correspondent B's mailing list without Correspondent B's knowledge; Correspondent A, however, cannot change or delete that mailing list. Correspondent A can examine the contents of Correspondent B's mailing list only if the processes used by both correspondents are running at the same node.
Programs written for a TRANSFER application reference various TRANSFER objects. These objects include: correspondents, depots, distribution lists, folders, items, packages, and profiles. Three of these objects are named:

- correspondents
- folders
- distribution lists

Correspondents, folders, and distribution lists are identified by TRANSFER simple names, either alone or in combination with each other. A simple name can have a maximum of 32 characters, and can include any of the following:

- hyphen (-) and underscore (_) (usually used as separators within names)
- letters A through Z (uppercase and lowercase are equivalent)
- digits 0 through 9

No other characters are permitted in a TRANSFER simple name. Embedded blanks and periods, which are regarded as delimiters, are not allowed.

Correspondents request a TRANSFER service through a request unit-of-work (UOW). Correspondents supply required names within 80-byte or 120-byte fields. These fields can contain leading blanks, but no character in a name can occupy the last byte; the last byte must contain a null (binary zero) character or a blank. A process can terminate a name at any point in the field by entering a null character after the last character in the name, in which case TRANSFER pads the trailing portion of the field with blanks.
TRANSFER Naming Conventions

TRANSFER returns a name to a process in a response UOW. The name is returned in an 80-byte or 120-byte field that is padded on the right with blanks.

As TRANSFER simple names enter the system, they are converted to an internal format that is used by the system software to resolve these names and manage the objects they represent.

Examples of TRANSFER simple names are:

BROWN

BROWN_JOE  (The underscore or hyphen character can be used; the two characters, however, are not treated as equivalent. The underscore might be preferable when names contain hyphens. The character selected should be used consistently.)

BROWN-JOE

MYPROC-2

123-5

CORRESPONDENT NAMES

A process makes a correspondent known to the system by supplying the name of the correspondent in a CREATE-DEPOT UOW to the TRANSFER interactive server, TISERV. This action is known as registration. A correspondent is registered at the node where the TISERV that services the registration request is running.

Once a correspondent is registered, any process can send a package to or act on behalf of that correspondent by supplying the correspondent name with the request for the desired task.

You can reference a correspondent anywhere in your program by entering the correspondent name in the following format:

correspondent-name [ @ [ node ] ]

where

correspondent-name

is a simple name that identifies the correspondent. A suggested format for naming correspondents is:

lastname-firstname-middleinitial
@ indicates that correspondent-name is a fully qualified name. JOE @, for example, is recognized immediately as representing the correspondent named JOE at the local node.

Embedded blanks are allowed between correspondent-name and @.

If @ is omitted, TRANSFER must use a search list and scan internal directories to fully resolve the correspondent name. A standard search list is built for each depot when the depot is created.

node

is the name of the network node where the correspondent is registered. Only alphanumeric characters can be used in the node name. You must omit the backslash (\) with which the name begins in GUARDIAN/EXPAND format. The node identified as \NY, for example, would be referenced as simply NY in your application program.

Embedded blanks are allowed between @ and node.

If node is omitted, TRANSFER assumes the local node.

When a process refers to a correspondent who is registered at the same node as the node at which the process is running, the node specification can be omitted. A process running on the system named \TM, for example, could address a correspondent named Parker-Jonathan @ TM simply by referencing Parker-Jonathan in the appropriate UOW.

Examples of correspondent names are:

- BRADLEY-ANNE a person at the local node
- CAMERON-ROBERT @ a person at the local node with the name fully qualified
- PROCESS-A @ TM a process at node \TM
- INVENTORY @ CORPR an application at node \CORPR
- SALES-MGR @ CORPR a person at node \CORPR
TRANSFER Naming Conventions

A process can address by name any correspondent in the network. A process, however, is subject to the following restrictions:

- A process can read or update only those profile records associated with the correspondent that the process presently represents.

- A process can create distribution lists or folders only for the correspondent that the process presently represents.

- A process can handle profile records for other correspondents at a node only if the process represents a correspondent with system administrator capability at that same node.

A process that adds recipients to a package or distribution list uses an extended form of correspondent name called a recipient name.

The recipient name can include a suffix enclosed in parentheses. TRANSFER carries the suffix within packages for use by agents or other subsystems in your application. The primary purpose of the suffix is to pass additional addressing information between Tandem networks and other networks.

The suffix is not considered part of the correspondent name and is not used in name resolution. The suffix does, however, determine whether a name is a duplicate; two names that differ only in their suffixes refer to the same depot, but are carried in packages as distinct names. If a package is addressed to GREEN (6634) and GREEN (6635), for example, both names are carried in the recipient list but the package is delivered only once; the package is delivered to the INBOX at the GREEN depot. Calls to agents at that depot, however, would take place once for each suffix.

A process supplies a recipient name in a 120-byte field. Leading blanks are permitted within this field. Characters in a name or suffix cannot extend beyond the 119th byte; the 120th byte must contain a null character (binary zero) or a blank. With the exception of the field length and suffix, the rules for recipient names are the same as those for other types of names.

Commas, single or double quotation marks, and nested parentheses are not allowed within the bounding parentheses of the suffix. The suffix can be separated from the correspondent name by one or more blanks. A character string must not follow the suffix.

TRANSFER stores a name containing a suffix as follows:

- converts all lowercase letters to uppercase

- removes all blanks immediately following the left parenthesis and immediately preceding the right parenthesis
• converts all other occurrences of multiple blanks into a single blank.

For example:

GREEN ( a suffix ) is stored as GREEN (A SUFFIX)

Examples of recipient names are:

CONNERS-ADAM (ADMIN)
OTHER-MAIL @ CORPR (FINANCE)
OTHER-MAIL @ CORPR (ACC-RECEIVABLE)

FOLDER NAMES

In addition to the special folders INBOX, OUTBOX, and WASTEBASKET maintained by TRANSFER, a process can define other folders for a depot. Folders can be used only by the depot owner. A process can do the following for the correspondent that the process represents:

• create or delete folders at the depot
• save items and packages in folders at the depot
• establish ordering criteria for folders so items and packages can be stored and retrieved in a specific sequence
• remove items and packages from folders at the depot
• request a list of folders belonging to the depot
• examine the contents of folders belonging to the depot.

Only a correspondent with system administrator capability at a node can create folders, delete folders, or obtain a list of folders that belong to another correspondent at that same node.

A process can reference an existing folder by entering the folder name in the following format.
TRANSFER Naming Conventions

[correspondent-name.]folder-name [ @ [ node ] ]

where

correspondent-name

is a simple name that identifies the correspondent. If
the correspondent name is omitted, TRANSFER searches for
the folder at the depot of the correspondent that the
process is representing.

folder-name

is a simple name that identifies the folder.

You cannot assign a folder the same name as a distribution
list belonging to the same depot because folders and
distribution list names are entered in the same format.

@ indicates that correspondent-name.folder-name is a fully
qualified name and requires no resolution by TRANSFER.

If @ is specified, the correspondent name is required and
must be included. If @ is omitted, TRANSFER is required
to fully resolve the name.

node

is the name of the network node where the correspondent is
registered. Only alphanumeric characters can be used in
the node name. You must omit the backslash (\) with which
the name begins in GUARDIAN/EXPAND format.

Embedded blanks are allowed between @ and node.

If node is omitted, TRANSFER assumes the local node.

Examples of folder names are:

LETTERS  a folder with the name LETTERS
in the depot of the
correspondent
TRANSFER Naming Conventions

WINKLER-FRANCIS-D.REPLIES @ a folder named REPLIES in the depot belonging to Francis D Winkler; a fully qualified folder name

TRACY-BILLNEWSLETTER @ TSY a folder named NEWSLETTER in the depot belonging to Bill Tracy at the node named \TSY; a fully qualified folder name.

DISTRIBUTION LIST NAMES

A process can define distribution lists for the correspondent that the process presently represents, and can add and delete members from those lists. A process representing a correspondent with system administrator capability can perform these operations with any lists at any depot at that correspondent's node.

A correspondent registered at a node can reference and display the contents of any distribution list at that same node. A system administrator registered at a node can create, read, alter, or delete any distribution list at that same node.

You can reference an existing distribution list in your program by entering the distribution list name in the following format:

\[correspondent-name.]distribution-list-name [ @ [ node ] ]

where

correspondent-name

is a simple name that identifies the correspondent who owns the list.

If the correspondent name is omitted, TRANSFER assumes the name of the correspondent currently represented by your process.

distribution-list-name

is a simple name that identifies the list.

You cannot assign a distribution list the same name as a folder belonging to the same depot because folders and distribution list names are entered in the same format.
TRANSFER Naming Conventions

@ indicates that correspondent-name.distribution-list-name is a fully qualified name that requires no resolution by TRANSFER.

If @ is specified, the correspondent name is required and must be included. If @ is omitted, TRANSFER is required to fully resolve the name.

node

is the name of the network node where the list is defined. Only alphanumeric characters can be used in the node name. You must omit the backslash (\) with which the name begins in GUARDIAN/EXPAND format.

Embedded blanks are allowed between @ and node.

If node is omitted, TRANSFER assumes the local node.

Examples of distribution list names are:

MY-LIST       a list with the simple name MY-LIST
SMITH-BOB.SALES a list containing the correspondent names of all local salespeople originated by a correspondent identified as SMITH-BOB
JAMES-DON.SALES @ NY a list containing the correspondent names of all salespeople working in the New York office
RAND-MARY.SALES @ CHI a list containing the correspondent names of all salespeople working in the Chicago office
TRANSFER Naming Conventions

JAMES-DON.SALES @ CHI a list containing the following
distribution list names:

SALES-MANAGER @ NY
JAMES-DON.SALES @ NY
RAND-MARY.SALES @ CHI

The first name is the name of an
individual correspondent. The last
two names are names of distribution
lists, illustrating the concept of
nested lists. All names are fully
qualified.

Any correspondent who knows the name of a list can use that list.
Notice that the members of a list need not all be located at the
same node. If a list contains the names of other lists, as in
the previous example, those lists need not have been created by
the same correspondent.

You can use a distribution list to associate a function with a
person. For example, you could establish a distribution list
named LIST.MANAGER-ACCOUNTING that had exactly one member named
SMITH-BOB. In this case, MANAGER-ACCOUNTING would be easier for
people to remember if they knew the function, but had little or
no contact with Bob Smith.

WILDCARD NAMES AND PATTERNS

In any context except the initial definition of a name, a process
can place an asterisk anywhere in a simple name to indicate that
any character or characters can appear. The simple name can be a
correspondent name, distribution list name, or folder name as
long as the abbreviated name identifies only one correspondent,
distribution list, or folder. This capability is known as using
wildcard characters in TRANSFER names. For example, *-Robert
identifies a correspondent whose first name is Robert.

If you place an asterisk at the end of a name, the process needs
to supply only enough characters to uniquely identify the name.
If Benson-Jill and Benson-Jonathan are both defined, Benson-Ji*
is sufficient to identify Benson-Jill unless another name also
begins with Benson-Ji.

You cannot use the asterisk if the resulting expression
identifies more than one name at the node; the name specified
must be unique. If, for example, two correspondents are defined
at the node with the last name Smith, the entry Smith-* is
ambiguous and results in a resolution error.
TRANSFER Naming Conventions

Using an asterisk as the first part of a name results in significant performance penalties. For example, resolving the name *-MARY is more costly than resolving the name SMITH-*.

A process can request a list of correspondent names, distribution list names, or folder names by entering a pattern for the listing. A pattern is similar to a wildcard name; asterisks are used to indicate that any character or characters (zero or more) can appear in that position of the name. A pattern, however, does not have to identify a unique name.

A process requesting a list of all correspondents at the node named CHIC with the last name of Jones can use the pattern

JONES-* @ CHIC

NODE DESIGNATION

If a correspondent at a node references a TRANSFER name defined at that same node, the correspondent need not enter the node designator. Specific ramifications of this rule are as follows:

• A new correspondent must be registered at the same node as the correspondent who requests the registration; therefore, a process can omit the node designation from the new correspondent name in the CREATE-DEPOT UOW. For example, LOUIS-JAMES @ TS can register BAKER-JON @ TS simply as BAKER-JON.

• A correspondent must initiate a session at the node where that correspondent name is registered; therefore, a process can omit the node designation from the correspondent name in any request to establish a session.

• During package delivery, TRANSFER processes at each recipient node resolve names and expand distribution lists that were defined at that node; therefore, a user process adding a name to a distribution list can omit the node designation if the name is defined at the same node as the distribution list. Note, however, that the user process must include the node designation if the name is defined at a different node.

NAME LENGTH RESTRICTION

TRANSFER converts the external format of object names into an internal format to resolve the names and manage the objects that they represent. The internal format includes the name of the node where the object is defined, the name of the TRANSFER name directory that contains the definition, and all other simple
names needed to uniquely identify the object. As an example, a folder identified externally as

USER.FOLDER @NODE

would be identified internally as

$\%\NODE$.ST.CORR.USER.FOLDER

where $\%\NODE$ indicates the name directory.
The internally expanded name must not exceed 79 characters.

To avoid having a simple name rejected because of a name length conflict, your application and its users should avoid excessively long names.

TRANSFER NAME DIRECTORY

The names of all correspondents, folders, and distribution lists in a TRANSFER application are defined in the TRANSFER name directory. This directory exists at each node, and identifies all named objects defined at the node. At node \NY, for example, the directory contains the names of all correspondents defined at the New York node.

The recommended name for identifying the name directory is $\%T.CORR$, which is also the default name supplied by Tandem. This same name must be used for the name directory at each node in the network. Differing names would imply more than one directory, but a TRANSFER system cannot communicate across directories.

For application development, you can establish and use an additional TRANSFER system with a different name directory and PATHWAY control file. This additional system could run in parallel with your production TRANSFER system; its name directory would be assigned a name other than $\%T.CORR$ through the methods noted in the TRANSFER System Management and Administration Guide. Notice, however, that a TRANSFER system using one name directory name cannot send packages to, or otherwise communicate with, another TRANSFER system using a different name directory name.
TRANSFER Naming Conventions

NAME SEARCHING

Two levels of objects are defined in the TRANSFER name directory. Correspondents are at the higher level. Folders and distribution lists are at the lower level.

This structure means that folders and distribution lists, effectively, exist only within depots. Within a depot, names must be unique; thus, a folder and a distribution list within the same depot cannot have the same name.

During name resolution, the list of objects at the lower level is searched before the list of objects at the higher level. During this search, an object name might match entries defined at both levels in the name directory. For example, a correspondent named ANNIE can have a distribution list named JOHN while a correspondent named JOHN also exists in the system. If ANNIE sends a message to JOHN, the distribution list name will be used because it is the first exact match found.

This searching algorithm is bypassed if a name is fully qualified. An address of JOHN @SYS immediately selects the correspondent and not the distribution list. Alternatively, an address of ANNIE.JOHN @SYS selects the distribution list; an address of ANNIE.JOHN also selects the distribution list because this name is resolved only at the lower level.

The presence of the @ character specifies that a name is fully qualified; by definition, the name is unique within the system. In the previous example, no other name exists anywhere that is exactly and entirely JOHN; the name ANNIE.JOHN is a name entirely different from JOHN, and represents an entirely different object.

If a wildcard character (*) occurs in the name, TRANSFER checks to determine if the name is ambiguous. Ambiguity exists if more than one possible candidate will resolve the name during the search at a particular level. When ambiguity between names exists at different levels, the candidate at the lower level is selected. When ambiguity exists at the same level and a unique entity is required, an error occurs. Using the previous example, TRANSFER would interpret the address JO* as the distribution list JOHN, assuming no other distribution list began with the characters JO. The fact that the next entry in the search, the correspondent JOHN, also matched JO* would be irrelevant.
Processes interface with TRANSFER by issuing requests to and receiving responses from the TRANSFER interactive server, TISERV.

TISERV handles requests one at a time, even when there are multiple openers, and performs the following functions:

- starts and terminates sessions
- services item and package handling requests, including those for creating, changing, and deleting items; composing, posting, and canceling packages; creating recipient lists for packages; and creating, scanning, and deleting folders
- handles administrative requests, such as those for creating and deleting depots; altering correspondent profiles and agent selection criteria for depots; and creating, altering, and deleting distribution lists.

Processes also interface with TRANSFER asynchronous requesters, TAREQs. When a package arrives at a depot, a TAREQ transmits a standard message to selected agents defined at the depot. The appropriate agents, in turn, can perform additional processing required by the application.

STARTING A TISERV PROCESS

Although most clients are written as SCREEN COBOL programs that execute within PATHWAY TCPs, clients can be written in FORTRAN, COBOL, or TAL. If you are writing a client in a language other than SCREEN COBOL, you can start a TISERV process for use by the client. You must observe the following rules.
Interfacing with TRANSFER

- TISERV must run under the same GUARDIAN user ID that was used to initialize the TRANSFER data base. If your client runs under a different user ID, create a copy of the TISERV program file owned by the TRANSFER user ID with the PROGID bit set in its label.

- Before starting a TISERV process, the name server process must be running.

- Before the TISERV process can do many kinds of useful work, the TRANSFER scheduler process must be running.

- When calling the GUARDIAN procedure NEWPROCESS, you should either omit the memory pages parameter or specify a value of zero. This causes the TISERV process to be created with its default virtual memory size.

TISERV interprets the startup message as follows:

default subvolume - ignored
IN file name - ignored
OUT file name - used to write serious error messages; these messages usually indicate a problem in the data base or the operational environment, not in the client
parameter string - ignored

Table 4-1 lists the parameters that TISERV accepts at startup time; none of the parameters are required.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Default Value</th>
<th>Recommended Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEBUGLOGFORMAT</td>
<td>FALSE</td>
<td>As desired</td>
<td>If FALSE: the debugging log will contain binary characters suitable for writing to an entry-sequenced disc file.</td>
</tr>
</tbody>
</table>
Table 4-1. TISERV Parameters at Startup Time (Continued)

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Default Value</th>
<th>Recommended Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEBUGLOGFORMAT (cont'd)</td>
<td></td>
<td></td>
<td>If TRUE: the debugging log will be formatted in ASCII characters suitable for writing to a terminal or printer.</td>
</tr>
<tr>
<td>DEBUGLOGLEVEL</td>
<td>3</td>
<td>As desired</td>
<td>If the PW-REPLY-CODE field of any response is greater than or equal to this value, the message and reply will be written to the debugging log. Specifying a value of zero causes all messages and replies to be logged.</td>
</tr>
<tr>
<td>DEBUGLOGRECSPEROPEN</td>
<td>20</td>
<td>As desired</td>
<td>After this many IPCs and their replies are written to the debug log, the log is closed and reopened.</td>
</tr>
<tr>
<td>HANGAROUND</td>
<td>FALSE</td>
<td>FALSE</td>
<td>If FALSE: TISERV stops after all requesters have closed it. If TRUE: TISERV never stops automatically.</td>
</tr>
</tbody>
</table>

**CAUTION**

When requesters and servers are both running in the PATHWAY environment, the HANGAROUND parameter for the PATHWAY server must be set to FALSE.

When a requester is running outside the PATHWAY environment and accessing a PATHWAY server, that PATHWAY server must be in its own server class so the HANGAROUND parameter can be set to TRUE; with the TRUE setting, the server remains available after it has been closed by the last opener.
Interfacing with TRANSFER

Table 4-1. TISERV Parameters at Startup Time (Continued)

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Default Value</th>
<th>Recommended Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDLESESSIONDELAY</td>
<td>24</td>
<td>24</td>
<td>Longest time, in hours, a session can be unused before TRANSFER automatically terminates it.</td>
</tr>
<tr>
<td>IOTIMEOUT</td>
<td>20</td>
<td>20</td>
<td>Longest time, in seconds, to wait for TRANSFER database I/O requests that are blocked due to other transactions.</td>
</tr>
<tr>
<td>ITEMIDCACHE</td>
<td>20</td>
<td>20</td>
<td>Number of ITEM IDs TISERV allocates at one time.</td>
</tr>
<tr>
<td>MAXLINKS</td>
<td>8</td>
<td>1</td>
<td>Number of simultaneous openers TISERV can support.</td>
</tr>
<tr>
<td>MAXREPLY</td>
<td>3000</td>
<td>As needed</td>
<td>Size of largest reply TISERV can generate.</td>
</tr>
<tr>
<td>MAXREQUEST</td>
<td>2400</td>
<td>As needed</td>
<td>Size of largest request TISERV can accept.</td>
</tr>
<tr>
<td>NAMESPACE</td>
<td>$T.CORR</td>
<td>As needed</td>
<td>Name of correspondent directory; value must be the same as was specified when TRANSFER was initialized.</td>
</tr>
</tbody>
</table>

TISERV accepts the DEBUGLOG parameter on the ASSIGN command at initialization time. The parameter has no default value. This parameter specifies the file to which a debugging log is written. The log contains a copy of some or all of the messages received and replied to by TISERV. The filename supplied via ASSIGN DEBUGLOG must be fully qualified; the system does not supply default values.
Interfacing with TRANSFER

If this parameter is omitted, no debugging log is written.

If this parameter refers to a disc file that does not exist, an entry-sequenced file will be created.

This parameter cannot refer to an EDIT file.

If the HANGAROUND parameter is FALSE, TISERV terminates normally after it is closed by all processes that opened it.

If TISERV encounters a fatal error, it attempts to write one or more messages to the OUT file (or the home terminal if the OUT file cannot be opened), and then ABENDs.

TISERV INTERFACE

Processes interface with TISERV through units-of-work (UOWs) issued within requests to TISERV. The format of requests issued by your program is the same, whether the request is issued by a SEND statement or by a WRITEREAD call. The data buffer for each request consists of the following:

- A request header that denotes the session on whose behalf the request is being made and that provides space for reply and return codes.
- One or more UOWs, each of which specifies a code for an operation plus any necessary parameters. For example, the operation ADD-RECIP would include the name of the recipient to be added.

The TISERV reply, which is written in the same format as the request to which it responds, consists of the following:

- A reply header, with reply and return codes supplied by TISERV.
- One or more response UOWs that return requested data to the correspondent. For example, the response to a CREATE-ITEM UOW request would include the item ID for the new item.

For each UOW in the request there is a corresponding response UOW in the overall reply. TISERV processes UOWs in the order in which they appear in the request. Thus, the response UOWs are returned in the same order as the corresponding UOWs in the request as illustrated in Figure 4-1.

For certain errors, however, no response UOWs are included. The number of UOWs in the reply is indicated by a field in the reply header.
Interfacing with TRANSFER

Request

<table>
<thead>
<tr>
<th>Request Header</th>
<th>Unit of Work 1</th>
<th>Unit of Work 2</th>
<th>Unit of Work 3</th>
<th>Unit of Work 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>operation</td>
<td>parameters</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

for example

<table>
<thead>
<tr>
<th>CREATE^ITEM</th>
<th>item type</th>
</tr>
</thead>
<tbody>
<tr>
<td>creates an item</td>
<td></td>
</tr>
</tbody>
</table>

Reply

<table>
<thead>
<tr>
<th>Reply Header</th>
<th>Reply to UOW 1</th>
<th>Reply to UOW 2</th>
<th>Reply to UOW 3</th>
<th>Reply to UOW 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>operation</td>
<td>return code</td>
<td>return data</td>
<td></td>
</tr>
</tbody>
</table>

for example

<table>
<thead>
<tr>
<th>CREATE^ITEM</th>
<th>okay</th>
<th>item id</th>
</tr>
</thead>
<tbody>
<tr>
<td>returns the id of the new item</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 4-1. TRANSFER Requests and Replies

Each UOW has a standard header that identifies the operation requested. Each response UOW has a return code that identifies the action taken in response to the corresponding request UOW.

A single SEND statement or WRITEREAD call can request multiple services as long as all of those services are for the same session. There is only one request header to identify the session. Refer to Section 6 for guidelines on the number of UOWs to include in an IPC and the number of IPCs to include in a TMF transaction.

Complete Data Definition Language (DDL) definitions of fields and structures used in TRANSFER interprocess messages are supplied as part of the standard software. Examples of their use appear in Sections 7 and 8.
Interfacing with TRANSFER

Request and Reply Headers

The format for request and reply headers is the same for all requests and replies. Within an application program, these headers are defined together as the interprocess communication (IPC) header. Your application must establish values for the IPC header fields that are transmitted in the request.

The format of the IPC header is shown by the following DDL definition:

```
DEF ipc-hdr.
  02 request-code TYPE BINARY 16.
    88 stop-on-warning VALUE -1.
    88 stop-on-err VALUE -2.
    88 do-all-uows VALUE -3.
  02 pw-reply-code TYPE BINARY 16
    REDEFINES REQUEST-CODE.
    88 all-uows-ok VALUE 0.
    88 uows-with-warning VALUE 1.
    88 uows-with-err VALUE 2.
    88 rqst-err VALUE 3.
  02 filler PIC X.
  02 version-code.
    03 letter PIC A.
    03 rev-number PIC 99.
  02 ipc-retn-code TYPE BINARY 16 VALUE 0.
    88 ipc-ok VALUE 0.
    88 invalid-version-code VALUE 1.
    88 invalid-session-id VALUE 2.
    88 service-denied VALUE 3.
    88 invalid-uow-header VALUE 4.
    88 rqst-too-long VALUE 5.
    88 reply-too-long VALUE 6.
    88 rqst-too-short VALUE 7.
    88 invalid-request-code VALUE 8.
    88 e-bad-transaction VALUE 4010.
    88 e-err-profile-file VALUE 4902.
    88 e-err-session-file VALUE 4904.
    88 e-err-itemdesc-file VALUE 4906.
    88 e-err-itemdata-file VALUE 4908.
    88 e-err-recip-file VALUE 4910.
    88 e-err-folder-file VALUE 4912.
    88 e-err-dlist-file VALUE 4914.
    88 e-err-ready-file VALUE 4916.
    88 e-err-time-file VALUE 4918.
    88 e-err-net-file VALUE 4920.
    88 e-err-inv-folder-file VALUE 4922.
    88 e-err-queue-file VALUE 4924.
    88 e-io-timeout VALUE 4990.
```

Interfacing with TRANSFER

```plaintext
88 e-waitmanager-unavail VALUE 6006
02 ipc-retn-code-detail TYPE BINARY 16 VALUE 0.
02 session-id.
  03 dummy PIC X(18).
02 uows-to-process TYPE BINARY 16 UNSIGNED.
02 uows-returned TYPE BINARY 16 UNSIGNED VALUE 0.
02 log-this-ipc TYPE BOOLEAN.
02 filler PIC X.
END.
```

Individual fields in the IPC header can contain the following information:

- **REQUEST-CODE**

In a request to TRANSFER, your application sets this field to indicate request processing conditions. In the request, this field always contains a value less than zero, as follows:

- **STOP-ON-WARNING** (-1) stops the processing if a UOW warning indication is encountered. Warnings imply successful completion of the UOW in which they occur.

- **STOP-ON-ERR** (-2) stops the processing if a UOW error occurs. Errors imply that the UOW was not processed successfully.

- **DO-ALL-UOWS** (-3) requests processing of all UOWs in the request. Processing halts only if a request error, as defined by the RQST-ERR value, or a system error external to TRANSFER occurs.

- **PW-REPLY-CODE**

In a reply from TRANSFER, the REQUEST-CODE field is redefined as the PW-REPLY-CODE field. This field contains a value that indicates request processing results. In the reply, this field always contains a value of zero or greater, as follows:

- **ALL-UOWS-OK** (0) indicates that all UOWs in the request were processed successfully.

- **UOWS-WITH-WARNING** (1) indicates that TRANSFER encountered warning indications in one or more UOWs. If STOP-ON-WARNING was not specified in the REQUEST-CODE field, all UOWs in the request have a corresponding response UOW in the reply. If STOP-ON-WARNING was specified, only those UOWs preceding and including the first with a warning indication have corresponding response UOWs.
Interfacing with TRANSFER

UOWS-WITH-ERR (2) indicates that TRANSFER detected errors in one or more UOWs. If neither STOP-ON-ERR nor STOP-ON-WARNING was specified in the REQUEST-CODE field, all UOWs in the request have a corresponding response UOW. If STOP-ON-ERR or STOP-ON-WARNING was specified, only those UOWs preceding and including the first with an error have corresponding response UOWs. If STOP-ON-ERR is specified in the request, the UOWs preceding the first with an error might return warning indications.

RQST-ERR (3) indicates that a request error occurred. This type of error typically indicates that something was wrong with the data in the IPC header and that the error is not specific to any particular UOW. In certain cases, the error might involve an individual UOW—for example, one with an invalid UOW header. When this error occurs, the number of response UOWs might be less than the number of UOWs in the request. For further information, your application should examine the IPC-RETN-CODE field.

CAUTION

If your application receives a value of 2 or 3 in this field, the transaction should be aborted, causing transaction backout. If the transaction is not backed out, consistency in the TRANSFER data base cannot be guaranteed.

• VERSION-CODE

In both a request to and a reply from TRANSFER, this field designates the version code for the IPC structure used. The version code is defined by Tandem and consists of a letter followed by a two-digit revision number.

The first release of TRANSFER was version code A01; new features were introduced in versions A02 and B00. If you use these new features, your application cannot run with previous versions of TRANSFER. You should set the version code in your IPC headers to the newest version of TRANSFER from which you used new features. TRANSFER will then return an INVALID-VERSION-CODE error in the IPC-RETN-CODE if you attempt to use your application with a version of TRANSFER that does not support all the new features you use.

IPCs established under the current version of TRANSFER will also be supported by subsequent versions of this product. You would not need to alter the IPCs used by your client when running under new versions of TRANSFER.
Interfacing with TRANSFER

- IPC-RETN-CODE

In a request to TRANSFER, this field is ignored. In a reply from TRANSFER, one of the following values appears:

IPC-OK (0) indicates that TRANSFER detected no errors in the IPC header. Warning indications or errors, however, might have been present in the individual UOWs in the request.

INVALID-VERSION-CODE (1) indicates that the request contained an IPC version code that could not be recognized by TRANSFER.

INVALID-SESSION-ID (2) indicates that the request contained a session ID that could not be recognized by TRANSFER. For example, the IPC might have referenced a session that no longer exists.

SERVICE-DENIED (3) is reserved for use by Tandem.

INVALID-UOW-HDR (4) indicates that an invalid UOW header appeared in your request. This could occur if the previous UOW specified the wrong size or if you omitted the value UW from the UOW header.

RQST-TOO-LONG (5) indicates that the request was too long; that is, the request contained more UOWs than the number specified in the UOWS-TO-PROCESS field, or contained more data than the TRANSFER buffer allowed.

REPLY-TOO-LONG (6) indicates that the reply was too long; the buffer space allotted for the reply was insufficient.

RQST-TOO-SHORT (7) indicates that the request was too short; that is, the request contained fewer UOWs than the number specified in the UOWS-TO-PROCESS field, or fewer bytes were sent than were expected.

INVALID-REQUEST-CODE (8) indicates that the REQUEST-CODE field contained an invalid entry.

E-BAD-TRANSACTION (4010) indicates that the request did not have a TMF transaction and attempted to perform an operation that required one, or that the transaction associated with the request is unusable, probably due to the failure of some component of the system or network. IPC-RETN-CODE-DETAIL contains the actual GUARDIAN file error code.
Interfacing with TRANSFER

E-ERR-PROFILE-FILE (4902) through E-ERR-INV-FOLDER-FILE (4922) indicate that an unexpected GUARDIAN file error occurred on one of the TRANSFER data base files. IPC-RETN-CODE-DETAIL contains the actual GUARDIAN file error code. If you receive one of these errors and the request was issued under a TMF transaction, you must abort the TMF transaction.

E-ERR-QUEUE-FILE (4924) indicates an error occurred on the Queue file. IPC-RETN-CODE-DETAIL contains the file code of the file on which the error occurred. If you receive this error and the request was issued under a TMF transaction, you must abort the TMF transaction; you should then retry the request under a new transaction.

E-IO-TIMEOUT (4990) indicates that a timeout occurred on I/O to the TRANSFER data base. This usually indicates a deadlock with another process. IPC-RETN-CODE-DETAIL contains the file code of the TRANSFER file on which the error occurred. If you receive this error and the request was issued under a TMF transaction, you must abort the TMF transaction; you should then retry the request under a new transaction.

E-WAITMANAGER-UNAVAIL (6006) indicates an error occurred when the Entry Manager was communicating with the Wait Manager.

- IPC-RETN-CODE-DETAIL

In a request to TRANSFER, this field is ignored. In a reply from TRANSFER, the meaning of this field depends on the value of IPC-RETN-CODE.

- SESSION-ID

In a request to TRANSFER, this field identifies the session of the correspondent on whose behalf the request is issued. In a reply from TRANSFER, this field shows the session ID that TRANSFER has assigned to the current session. When your application requests session initiation, this field must be set to binary zeros in the request; in subsequent requests, you must move the session ID returned by TRANSFER to this field.

- UOWS-TO-PROCESS

In a request, this field indicates the number of UOWs transmitted with the request. In a reply, TRANSFER echoes the value in the request.
Interfacing with TRANSFER

- **UOWS-RETURNED**

  In a request, this field is ignored. In a reply, TRANSFER indicates the number of response UOWs returned to your application.

- **LOG-THIS-IPC**

  In a request, this field specifies whether the request and its reply are logged (Y) or not logged (N). If the field is set to N, the depot criteria or the TRANSFER software might override your request and log the IPC in some cases.

**UOW Operations**

Processes communicate with TISERV by issuing requests that contain UOWs. UOWs are organized into the following functional areas:

- session control
- item handling (including whole item, record handling, and item tree operations)
- package handling (including submittal preparation, recipient list definition, package delivery, and package receipt)
- folder manipulation
- TRANSFER configuration inquiry
- administration (including session, depot, distribution list, and name management)

This section presents DDL definitions that are common to various functional areas; for example, the discussion of record handling functions includes the DDL definitions for significant fields shared among the UOWs that are common to that area.

For programs that are written in languages other than COBOL and SCREEN COBOL, fields defined as PIC 9(4) COMP are limited to values in the range of 0 through 9999. In TAL, for example, this field would have a type of INT, but entering a value outside the range would return an error.

Two DDL definitions are shared among several functional areas: BOOLEAN and UPDATE-CONTROL.
Interfacing with TRANSFER

- **BOOLEAN** - This definition is used to assign a data type of BOOLEAN to fields containing data that is either logically true (yes, represented by Y or y) or false (no, represented by N or n).

The DDL definition for BOOLEAN is:

```
DEF boolean PIC A.
```

When UOWs have OPTIONS fields defined as BOOLEAN, the BOOLEAN fields are checked for validity before any other fields in the UOW. Input UOW fields are not necessarily checked sequentially for validity. For example: if invalid values were entered in both the ORDERING-DISCIPLINE and ALLOW-DUPLICATES fields in the CREATE-FOLDER-B00 UOW, the error 4051 (E-MUST-BE-YN) for the ALLOW-DUPLICATES would be returned.

- **UPDATE-CONTROL** - This definition applies to a field used in UOWs that update records maintained under TRANSFER, such as those records comprising item descriptions. This definition is used to ensure that updating is accomplished in an orderly way.

The DDL definition for UPDATE-CONTROL is:

```
DEF update-control PIC S9(4) COMP.
```

When your process gains access to a record prior to updating it, you need some guarantee that another process will not modify that record before your process completes its own updating operations. To serve multiple users, TRANSFER cannot keep a record locked to prevent a multiple update; instead, TRANSFER must release the record locks as soon as it returns the record to your process. To resolve this conflict between its own needs and those of your process, TRANSFER uses the UPDATE-CONTROL field as follows.

1. Each time TISERV completes a record update, it increments the value of an update control count in that record.

2. When your process requests access to a record, TISERV returns the current update control count in the response UOW.

3. When your process modifies a record and subsequently requests update to that record, TISERV checks the value in the UPDATE-CONTROL field against the current update control count of the record. If these two values match, the update is allowed to proceed. If the values do not match, this indicates that another update took place while your process was accessing the record, and TISERV returns an error indication to your process; your process should retry the access and update operations in this case.
Interfacing with TRANSFER

SESSION CONTROL. Before a process can transmit requests, the process must establish communication with TISERV by initiating a session with the START-SESSION UOW. When interaction with TISERV is no longer required, the process terminates this communication by issuing an END-SESSION UOW.

These UOWs involve three fields of particular significance: SESSION-ID, CORR-NAME, and PASSWORD.

• SESSION-ID - When you issue the START-SESSION UOW, set the SESSION-ID field in the IPC header to binary zero; when TISERV responds with the response UOW, it returns in this field a session ID that uniquely identifies your session at the node. This session ID must appear in the IPC header of every subsequent request that the process makes for this session. When the process ends the session, it relinquishes the session ID.

The DDL definition for SESSION-ID is:

```
DEF session-id.
  02 dummy PIC X(18).
END.
```

SESSION-ID will contain unprintable characters. Any attempt to display the session ID will have unpredictable results.

• CORR-NAME - The CORR-NAME field identifies the name of the correspondent that the process represents. The field contains either a fully qualified or partially qualified correspondent name.

The DDL definition for CORR-NAME is:

```
DEF corr-name PIC X(80).
```

• PASSWORD - The PASSWORD field contains the password that the correspondent must supply in order to initiate a session.

The DDL definition for PASSWORD is:

```
DEF password PIC X(16).
```

ITEM HANDLING. Item handling involves three general types of operations:

- defining individual items (whole item operations)
- appending records to and deleting records from items (record handling operations)
attaching and detaching items as components of other items (item tree operations).

Whole Item Operations. Each item is made up of an item descriptor and one or more data records. These data records can contain text or other kinds of application data.

The item descriptor contains:
- a network-unique ID of the item
- a count of the number of component items pointed to by the first item; if the number is zero, a list of component items does not exist
- a list of item IDs for the component items
- various fields indicating the current status of the given item
- the current update control count for the item, the creation date, and the name of the correspondent who created the item
- an item type, which is a numeric value defined by the application; the item type should not be confused with the record type, as used in record handling operations.

If the item is a package, the item descriptor also contains:
- the date that the package was submitted for delivery
- the earliest and latest delivery dates that define the delivery window
- the optional expiration date for the package
- the package priority and delivery control information.

A complete listing of all information in the item descriptor appears in the discussion of the GET-ITEM-DESCRIPTION UOW in Section 5.

A process creates an item by issuing a CREATE-ITEM UOW. In response, TISERV assigns the item ID and records the creation date and the correspondent name, item type, and other information related specifically to this item in the item descriptor.
Interfacing with TRANSFER

A process accesses all information in the item descriptor by issuing a GET-ITEM-DESCR UOW. A process can make a duplicate copy of the item, with its own unique item ID, by issuing a COPY-ITEM UOW. You might want to do this to add records to an item where the original item is unalterable.

The item ID and item type are passed to and from TISERV in the ITEM-ID and ITEM-TYPE fields.

• ITEM-ID - The ITEM-ID field identifies an item.

The DDL definition for ITEM-ID is:

```
DEF item-id.
    02 dummy PIC X(12).
END.
```

ITEM-ID will contain unprintable characters. Any attempt to display the item ID will have unpredictable results.

• ITEM-TYPE - The ITEM-TYPE field categorizes an item.

The DDL definition for ITEM-TYPE is:

```
DEF item-type PIC 9(4) COMP.
```

ITEM-TYPE is a field that can be used by the application to categorize items by type. This field can have a value from 1000 through 9999. The values 100 through 999 are assigned for use by Tandem. Refer to Appendix B for additional information.

Record Handling Operations. A process can perform the following record handling operations:

- define and add records of various types to an item (ADD-ITEM-REC UOW)
- access existing records in an item (GET-ITEM-REC UOW)
- delete records (DELETE-ITEM-REC UOW).

In these UOWs, the ITEM-KEY field is vitally significant. When a process accesses records in sequence, the ITEM-KEY field determines the starting point for retrieval.

• ITEM-KEY - The ITEM-KEY field identifies the item to which the record belongs or will belong, and the specific record itself.
Interfacing with TRANSFER

The DDL definition for ITEM-KEY is:

```
DEF item-key.
  02 item-id.
    03 dummy PIC X(12).
  02 rec-type PIC 9(4) COMP.
  02 rec-seq-num PIC 9(4) COMP.
END.
```

ITEM-ID - The ITEM-ID field specifies the item to be accessed.

REC-TYPE - The REC-TYPE field describes what kind of record is being accessed; this allows you to access records by category. This field can have a value from 0 through 9999.

The values 100 through 999 are assigned for use by Tandem application packages, such as T/MAIL. You can use these values in the REC-TYPE field, but they should have the same meaning for your application as they do for such software. These record types include subject text, forward and response text, and unformatted ASCII text; refer to Appendix B for values used in the REC-TYPE field. An item can contain records of many different types and multiple records of the same type.

REC-SEQ-NUM - The REC-SEQ-NUM field indicates the specific record to be accessed. The field provides unique key values for multiple records having the same item ID and record type.

**Item Tree Operations.** A process can attach one item to another in a parent-component relationship. In fact, a process can arrange for a component item to have components, each of which in turn can have components; this allows an entire tree of items to be created. The number of levels of nesting, however, can affect the performance of your application.

A process can perform the following item tree operations:

- attach one item to another (ATTACH-COMPNT-A01 UOW)
- obtain the item IDs of component items belonging to a parent (GET-ITEM-COMPNT-A01 UOW)
- detach a component from a parent (DETACH-COMPNT UOW).
Interfacing with TRANSFER

The ATTACH-COMPNT-A01 UOW includes a COMPNT-TYPE field that is returned with the component by the GET-ITEM-COMPNT-A01 UOW.

- COMPNT-TYPE - The COMPNT-TYPE field is a numeric value that TRANSFER saves with a component but does not interpret.

  The DDL definition for COMPNT-TYPE is:

  `DEF compnt-type PIC 9(4) COMP.`

  The COMPNT-TYPE field can be used by the application as a code to indicate why the component was attached. The field can have a value from 0 through 9999; values 100 through 999 are reserved for Tandem. Refer to Appendix B for additional information.

PACKAGE HANDLING. Package handling involves the following activities:

- submittal preparation
- recipient list definition
- package delivery
- package receiving operations

Submittal Preparation. Before items can be delivered from one correspondent to another, they must be assembled as packages. Each package includes a package header item, which consists of zero or more records or components items, and a list of recipients.

The package header is a special kind of item, designated by the IS-PKG-HDR field of a CREATE-ITEM UOW. The item descriptor of a package header item indicates who is sending the package, who is receiving the package, and when the package was posted. Since a package can only be altered or submitted by its creator, the sender of a package is always the creator.

The package header can contain text records and item IDs of components. If the components list includes another package header item, the result is a package nested in another package.
The item descriptor of a package header also contains the following information that your application can supply: timestamps, priority, agent selector, and certification.

The simplest package is a package header with no components. A package header item, like any other item, can contain data records. For example, the package header might include a data record that contains a package sequence number.

A process builds a package by creating the package header item and making separate requests to add recipients, component items, and delivery parameters.

A process can perform the following operations:

- access information in the item descriptor (GET-ITEM-DESCR UOW)
- update fields in the item descriptor (ALTER-ITEM-DESCR UOW).

Four fields of principal interest are: AGENT-SELECTOR, APPLIC-ID, PRIORITY, and DATE-TIME.

- AGENT-SELECTOR and APPLIC-ID - The AGENT-SELECTOR and APPLIC-ID fields are used as a basis for agent selection, and their contents and meaning are defined by your application. After delivering a package, TRANSFER examines these two fields to determine what agent to invoke. An agent, in fact, can be configured to react to several AGENT-SELECTOR and APPLIC-ID values.

The definitions for AGENT-SELECTOR and APPLIC-ID are:

```plaintext
DEF agent-selector PIC 9(4) COMP.
DEF applic-id PIC 9(4) COMP.
```

The following APPLIC-ID values are reserved for Tandem use:

- 100-499 Tandem clients; ID 100 indicates a TAREQ, and ID 111 is T/Mail.
- 500-999 Tandem agents; ID 500 is the VACATION agent.
Interfacing with TRANSFER

- PRIORITY - The PRIORITY field determines the priority at which the package will be sent, allowing you to classify a package according to its urgency. The field contains an unsigned value ranging from 0 (lowest priority) through 199 (highest priority). Packages are sent in decreasing order of numeric priority; when two or more packages are assigned the same numeric priority, the packages are sent in order of submittal date.

Priority overrules submission time in determining when a package is sent; a high-priority package goes before a package of low priority, even if the low-priority package was submitted earlier. Assigning a high priority to new packages might prevent older packages from being delivered within the requested timeframe.

The DDL definition for PRIORITY is:

```
DEF priority PIC 9(3) COMP.
```

- DATE-TIME - The DATE-TIME field governs the format of various dates and times that appear in the package header item.

The DDL definition for DATE-TIME is:

```
DEF date-time.
  02 year PIC 9(4).
  02 month PIC 9(2).
  02 day-of-month PIC 9(2).
  02 hour PIC 9(2).
  02 minute PIC 9(2).
  02 second PIC 9(2).
END.
```

The DATE-TIME field can have values in the following ranges:

- YEAR 1975 through 2099
- MONTH 1 through 12
- DAY-OF-MONTH 1 through 31
- HOUR 0 through 23
- MINUTE and SECOND 0 through 59

In a reply, TRANSFER returns a DATE-TIME of all zeros for a time that has not been established; for example, the submission time of a package that has not been submitted.

**Recipient List Definition.** The recipient list contains the names of correspondents and distribution lists that are to receive the package. A process can perform the following operations for these lists:

- add names to a recipient list (ADD-RECIPI UOW)
- obtain the names of currently defined recipients (GET-RECIP-REC UOW)

- delete recipients from a distribution list (DELETE-RECIP UOW).

The ADD-RECIP and DELETE-RECIP UOWs both transmit the RECIP-NAME field and the contents of the RECIP-TYPE field. The GET-RECIP-REC UOW transmits a RECIP-KEY field.

- RECIP-NAME - The RECIP-NAME field identifies the recipient to be added or deleted. Names of correspondents in this field can include suffixes. The resolution of names and the expansion of distribution lists appearing in this field are described under the ADD-RECIP UOW in Section 5.

The DDL definition for RECIP-NAME is:

```ddl
DEF recip-name PIC X(120).
```

- RECIP-TYPE - THE RECIP-TYPE field is used to categorize various kinds of recipients. This field, essentially, explains why a recipient is receiving the package.

The DDL definition for RECIP-TYPE is:

```ddl
DEF recip-type PIC 9(4) COMP.
```

The RECIP-TYPE field can have a value from 0 through 9999. Values 100 through 999, however, are assigned for use by Tandem and might assume particular meanings when interfacing with software, such as T/MAIL, that is supplied by Tandem.

- RECIP-KEY - The RECIP-KEY field is used as a key in selecting the recipient names returned by the GET-RECIP-REC UOW. The field references both the package item and the name of one of the recipients.

The DDL definition for RECIP-KEY is:

```ddl
DEF recip-key.
  02 item-id.
    03 dummy PIC X(12).
  02 recip-name PIX X(120).
END.
```

Package Delivery. A process posts a package for delivery by issuing a SUBMIT-PKG UOW. A process cancels the delivery of a package by issuing a CANCEL-PKG UOW.
Interfacing with TRANSFER

Package Receiving. A process acknowledges receipt of a package by issuing an ACK-RECEIPT UOW. This operation is recommended whenever a client retrieves a package from the INBOX folder.

FOLDER MANIPULATION. Folder manipulation involves the following operations:

- save items in folders according to the ordering criteria established for the folder (SAVE-ITEM, SAVE-ITEM-B00, and SAVE-ITEM-BY-KEY UOWs)
- modify the ordering criteria for a folder (ALTER-FOLDER-ORDER UOW)
- scan the contents of a folder (SCAN-FOLDER, SCAN-FOLDER-B00, and SCAN-FOLDER-BY-KEY UOWs)
- remove an item from a folder (UNSAVE-ITEM UOW)
- determine the folders in which a particular item is saved (WHERE-SAVED UOW)
- determine the ordering criteria for a folder (GET-FOLDER-ORDER UOW).

A folder can contain both packages and individual items. Conversely, the same item or package can reside in more than one folder. An item cannot be stored in one folder more than once.

Folder manipulation UOWs either transmit or return the contents of the FOLDER-NAME field.

- FOLDER-NAME - The FOLDER-NAME field is the name of a folder.

The DDL definition for FOLDER-NAME is:

```
DEF folder-name PIC X(80).
```

On input, you can use a partially qualified name, including wildcard characters. In the response, the fully qualified name is returned.

TRANSFER CONFIGURATION INQUIRY. A process obtains information from the TRANSFER name configuration directory by issuing a GET-CONFIG-NAME UOW.
Interfacing with TRANSFER

ADMINISTRATION. A process requests various administrative operations, such as the management of depots, distribution lists, or TRANSFER names, by issuing administrative UOWs. Some of these UOWs can only be issued by processes representing correspondents with system administrator capability.

Many administrative UOWs include one or more fields that contain a correspondent name. These fields identify the correspondent that the calling process represents. In cases where the DDL for these fields includes a VALUE SPACES clause, only system administrators can enter correspondent names other than their own; users who are not system administrators must enter their own names or leave the field blank.

Session Management. A process can obtain the names of correspondents with currently active sessions by issuing a GET-NEXT-SESSION UOW. If you are writing your own administrative client, you can use this UOW in connection with terminating sessions, shutting down TRANSFER, and monitoring the load on the TRANSFER system. This UOW requires the system administrator capability.

Depot Management. A correspondent depot is established when that correspondent name is registered with TRANSFER. Every correspondent has precisely one depot and that depot has a network-unique identity. The depot contains profiles, folders, and distribution lists. Agents are also associated with depots.

A process creates a depot, and consequently registers a correspondent who owns that depot, by issuing a CREATE-DEPOT UOW. A process deletes a depot by issuing a DELETE-DEPOT UOW. Both of these UOWs require the system administrator capability.

Several UOWs pertain to the management of profiles. A process can do the following:

- obtain one or more data elements from a depot profile (GET-PROFILE-ELEM UOW)
- change data elements in a profile (ALTER-PROFILE-ELEM UOW)
- read user-maintained profile records (READ-PROFILE-REC UOW)
- write user-maintained profile records (WRITE-PROFILE-REC UOW)
- delete profile records (DELETE-PROFILE-REC UOW).
Interfacing with TRANSFER

Two UOWs are involved in managing folders: the CREATE-FOLDER UOW that creates a folder, and the DELETE-FOLDER UOW that deletes a folder. A folder must be created before an item or package can be saved in it.

In the area of agents defined at a depot, the GET-AGENT-SELECT UOW returns to a process the current selection criteria for a particular agent. The ALTER-AGENT-SELECT UOW defines, alters, or deletes this criteria.

Distribution List Management. Five UOWs are concerned with the management of distribution lists. A process can do the following:

- create a distribution list (CREATE-DLIST UOW)
- delete the distribution list (DELETE-DLIST UOW)
- add a new member to a distribution list (ADD-MEMBER UOW)
- delete a member (DELETE-MEMBER UOW)
- read a distribution list (READ-NEXT-MEMBER UOW).

A distribution list must be created before members can be added to it.

Name Management. The READ-NEXT-NAME UOW selectively reads the contents of the TRANSFER name directory, which defines all names known to TRANSFER. The READ-NEXT-NAME UOW can be used, for example, to read all folder names defined.
UOW Summary Table

All TISERV UOWs and the functions they perform are summarized in Table 4-2.

Table 4-2. TISERV UOW Summary

<table>
<thead>
<tr>
<th>UOW</th>
<th>UOW Code</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>END-SESSION</td>
<td>102</td>
<td>End a session in progress on behalf of a correspondent.</td>
</tr>
<tr>
<td>START-SESSION</td>
<td>101</td>
<td>Begin a session on behalf of a correspondent.</td>
</tr>
</tbody>
</table>

Session Control UOWs

These UOWs are concerned with the initiation and termination of sessions.

Item Handling UOWs

The item handling UOWs are used for creating and manipulating items and their components.

- Whole Item UOWs

These UOWs are used for defining individual items.

<table>
<thead>
<tr>
<th>UOW</th>
<th>UOW Code</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>COPY-ITEM</td>
<td>107</td>
<td>Make a duplicate copy of an item.</td>
</tr>
<tr>
<td>CREATE-ITEM</td>
<td>103</td>
<td>Create an item.</td>
</tr>
<tr>
<td>GET-ITEM-DESCR</td>
<td>122</td>
<td>Return descriptor fields for an item.</td>
</tr>
</tbody>
</table>
Interfacing with TRANSFER

Table 4-2. TISERV UOW Summary (Continued)

- Record Handling UOWs

These UOWs are used for appending records to items or deleting them from items.

<table>
<thead>
<tr>
<th>UOW</th>
<th>UOW Code</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD-ITEM-REC</td>
<td>104</td>
<td>Add a data record to an item.</td>
</tr>
<tr>
<td>DELETE-ITEM-REC</td>
<td>105</td>
<td>Delete a data record from an item.</td>
</tr>
<tr>
<td>GET-ITEM-REC</td>
<td>125</td>
<td>Retrieve data records from an item.</td>
</tr>
</tbody>
</table>

- Item Tree UOWs

These UOWs are involved in attaching items as components to other items, and detaching these components as well.

<table>
<thead>
<tr>
<th>UOW</th>
<th>UOW Code</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTACH-COMPNT-A01</td>
<td>136</td>
<td>Attach one item as a component of another.</td>
</tr>
<tr>
<td>DETACH-COMPNT</td>
<td>113</td>
<td>Detach one item from another.</td>
</tr>
<tr>
<td>GET-ITEM-COMPNT-A01</td>
<td>137</td>
<td>Return a list of component items within an item to your application.</td>
</tr>
</tbody>
</table>

Package Handling UOWs

The UOWs that handle packages are divided into four categories: submittal preparation, recipient list definition, package delivery, and package receipt UOWs.
Table 4-2. TISERV UOW Summary (Continued)

- **Submittal Preparation UOW**
  
  This UOW is concerned with operations that prepare packages for submittal (exclusive of the recipient list).

<table>
<thead>
<tr>
<th>UOW</th>
<th>Code</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTER-ITEM-DESCR</td>
<td>116</td>
<td>Update application-controlled fields in an item descriptor.</td>
</tr>
</tbody>
</table>

- **Recipient List Definition UOWs**
  
  These UOWs are concerned with defining and verifying recipients for packages.

<table>
<thead>
<tr>
<th>UOW</th>
<th>Code</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD-RECIP</td>
<td>114</td>
<td>Add a new recipient to the recipient list for a package.</td>
</tr>
<tr>
<td>DELETE-RECIP</td>
<td>115</td>
<td>Delete a recipient from the recipient list for a package.</td>
</tr>
<tr>
<td>GET-RECIP-REC</td>
<td>126</td>
<td>Retrieve recipient records from a package recipient list.</td>
</tr>
</tbody>
</table>

- **Package Delivery UOWs**
  
  These UOWs are concerned directly with the delivery of packages.

<table>
<thead>
<tr>
<th>UOW</th>
<th>Code</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CANCEL-PKG</td>
<td>118</td>
<td>Cancel delivery of a package.</td>
</tr>
<tr>
<td>SUBMIT-PKG</td>
<td>117</td>
<td>Submit (post) a package for delivery.</td>
</tr>
</tbody>
</table>
Interfacing with TRANSFER

Table 4-2. TISERV UOW Summary (Continued)

<table>
<thead>
<tr>
<th>UOW</th>
<th>UOW Code</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACK-RECEIPT</td>
<td>131</td>
<td>Acknowledge receipt of a package.</td>
</tr>
</tbody>
</table>

Folder Manipulating UOWs

These UOWs are used for the maintenance of folders.

<table>
<thead>
<tr>
<th>UOW</th>
<th>UOW Code</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAVE-ITEM</td>
<td>110</td>
<td>Save an item in a folder.</td>
</tr>
<tr>
<td>SAVE-ITEM-B00</td>
<td>141</td>
<td>Save an item in a folder that has any ordering discipline except APPLIC-DEFINED with an option to include an unsave time.</td>
</tr>
<tr>
<td>SAVE-ITEM-BY-KEY</td>
<td>139</td>
<td>Save an item in a folder that has an ordering discipline of APPLIC-DEFINED.</td>
</tr>
<tr>
<td>SCAN-FOLDER</td>
<td>120</td>
<td>Return the ID of items in a particular folder.</td>
</tr>
<tr>
<td>SCAN-FOLDER-B00</td>
<td>142</td>
<td>Return the ID and unsave time of items in a particular folder.</td>
</tr>
</tbody>
</table>
Table 4-2. TISERV UOW Summary (Continued)

Folder Manipulating UOWs (continued)

<table>
<thead>
<tr>
<th>UOW</th>
<th>Code</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCAN-FOLDER-BY-KEY</td>
<td>140</td>
<td>Return the ID, item type, unsave time, and the corresponding ordering key and key length of items saved in folders with APPLIC-DEFINED ordering.</td>
</tr>
<tr>
<td>UNSAVE-ITEM</td>
<td>111</td>
<td>Remove an item from a folder.</td>
</tr>
<tr>
<td>WHERE-SAVED</td>
<td>121</td>
<td>Return the names of folders in which a given item is saved.</td>
</tr>
</tbody>
</table>

TRANSFER Configuration Inquiry UOW

This UOW returns information from the TRANSFER name configuration directory.

<table>
<thead>
<tr>
<th>UOW</th>
<th>Code</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET-CONFIG-NAME</td>
<td>132</td>
<td>Return the configured name for the TRANSFER file, process, or other special entity, as recorded in the TRANSFER name directory.</td>
</tr>
</tbody>
</table>

Administrative UOWs

The administrative UOWs are typically used by processes to perform various administrative functions. These UOWs are divided into the following categories: session management, depot management, distribution list management, and name management UOWs. UOWs that can only be issued by processes representing users with system administrator capability are noted.
Table 4-2. TISERV UOW Summary (Continued)

- Session Management UOW

This UOW is used in connection with forcefully terminating sessions, shutting down TRANSFER, and monitoring the load on the TRANSFER system.

<table>
<thead>
<tr>
<th>UOW</th>
<th>Code</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET-NEXT-SESSION</td>
<td>200</td>
<td>Return the names of correspondents with active sessions to your application. (Requires system administrator capability.)</td>
</tr>
</tbody>
</table>

- Depot Management UOWs

These UOWs are concerned with the management of depots and the objects defined as essential components of depots: profiles, folders, and agents.

**Depot UOWs**

These UOWs are concerned with depots themselves.

<table>
<thead>
<tr>
<th>UOW</th>
<th>Code</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREATE-DEPOT</td>
<td>201</td>
<td>Create a depot. (Requires system administrator capability.)</td>
</tr>
<tr>
<td>DELETE-DEPOT</td>
<td>202</td>
<td>Delete a depot. (Requires system administrator capability.)</td>
</tr>
</tbody>
</table>
Table 4-2. TISERV UOW Summary (Continued)

Profile UOWs
These UOWs are concerned with profiles.

<table>
<thead>
<tr>
<th>UOW</th>
<th>UOW Code</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTER-PROFILE-ELEM</td>
<td>204</td>
<td>Modify one or more data elements in a profile.</td>
</tr>
<tr>
<td>DELETE-PROFILE-REC</td>
<td>214</td>
<td>Delete a record in a depot profile.</td>
</tr>
<tr>
<td>GET-PROFILE-ELEM</td>
<td>203</td>
<td>Return one or more data elements from a profile.</td>
</tr>
<tr>
<td>READ-PROFILE-REC</td>
<td>212</td>
<td>Read a record from a depot profile.</td>
</tr>
<tr>
<td>READ-PROF-REC-A02</td>
<td>231</td>
<td>Read one or more records from a depot profile.</td>
</tr>
<tr>
<td>WRITE-PROFILE-REC</td>
<td>213</td>
<td>Write or update a record in a depot profile.</td>
</tr>
</tbody>
</table>

Folder UOWs
These UOWs are concerned with folder management.

<table>
<thead>
<tr>
<th>UOW</th>
<th>UOW Code</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTER-FOLDER-ORDER</td>
<td>234</td>
<td>Modify the ordering criteria for an existing folder.</td>
</tr>
<tr>
<td>CREATE-FOLDER</td>
<td>227</td>
<td>Create a folder.</td>
</tr>
<tr>
<td>CREATE-FOLDER-B00</td>
<td>232</td>
<td>Create a folder with a specified ordering criteria.</td>
</tr>
<tr>
<td>DELETE-FOLDER</td>
<td>230</td>
<td>Delete a folder.</td>
</tr>
<tr>
<td>GET-FOLDER-ORDER</td>
<td>233</td>
<td>Retrieve the ordering criteria for a folder.</td>
</tr>
</tbody>
</table>
Interfacing with TRANSFER

Table 4-2. TISERV UOW Summary (Continued)

Agent UOWs

These UOWs are concerned with agents.

<table>
<thead>
<tr>
<th>UOW</th>
<th>UOW Code</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTER-AGENT-SELECT</td>
<td>208</td>
<td>Create, alter, or delete the selection criteria for an agent.</td>
</tr>
<tr>
<td>GET-AGENT-SELECT</td>
<td>207</td>
<td>Return the selection criteria for an agent to your application.</td>
</tr>
</tbody>
</table>

- Distribution List Management UOWs

These UOWs are concerned with creation, deletion, and maintenance of distribution lists.

<table>
<thead>
<tr>
<th>UOW</th>
<th>UOW Code</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD-MEMBER</td>
<td>220</td>
<td>Add a new member to a distribution list.</td>
</tr>
<tr>
<td>CREATE-DLIST</td>
<td>217</td>
<td>Create a distribution list.</td>
</tr>
<tr>
<td>DELETE-DLIST</td>
<td>218</td>
<td>Delete a distribution list.</td>
</tr>
<tr>
<td>DELETE-MEMBER</td>
<td>221</td>
<td>Delete a member from a distribution list.</td>
</tr>
<tr>
<td>READ-NEXT-MEMBER</td>
<td>219</td>
<td>Return the names of members of a distribution list.</td>
</tr>
</tbody>
</table>
Interfacing with TRANSFER

Table 4-2. TISERV UOW Summary (Continued)

<table>
<thead>
<tr>
<th>UOW</th>
<th>Code</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>READ-NEXT-NAME</td>
<td>224</td>
<td>Read the contents of the TRANSFER name directory.</td>
</tr>
</tbody>
</table>

The actual delivery of a package to a depot is handled by any of several TRANSFER asynchronous requesters (TAREQs). As part of the delivery operation, the TAREQ can optionally invoke one or more user-written agents that perform additional processing related to the delivery. TAREQs are composed of SCREEN COBOL programs and, therefore, they interface with agents that are implemented either as SCREEN COBOL requester programs or as PATHWAY servers.

Agent Selection

Upon package delivery, the responsible TAREQ can select agents for execution by matching delivery parameters that accompany the package against agent selection criteria defined in profile records at the recipient depot. The TAREQ passes the delivery parameters as part of an agent notification message. Your application establishes the agent selection criteria through the ALTER-AGENT-SELECT UOW or the TRANSFER/ADMIN client.

When the package arrives at the depot, the TAREQ selects the appropriate agents as described in the ALTER-AGENT-SELECT UOW in Section 5.

In selecting agents, the TAREQ sequentially searches the agent selection criteria in the profiles for the depot. During this search, the TAREQ invokes, in sequence, each agent whose selection criteria is satisfied by the delivery parameters in the notification message. The TAREQ continues this process until no further qualified agents are encountered or until an agent requests the TAREQ to discontinue agent selection.
Interfacing with TRANSFER

If an agent selection record for a selected agent specifies that a session should be started when the agent is invoked, the TAREQ starts a session and obtains a session ID from TISERV. The TAREQ includes the session ID in the agent notification message; this allows an agent that can be invoked by more than one depot to access depots for which it is configured without knowing the password for the depot.

When more than one agent requiring a session is selected, the TAREQ assigns each agent an individual session. As each agent returns control, the TAREQ ends the session on the behalf of the agent. When two agents are involved, for example, the TAREQ might begin by starting a session for Agent A, and then invoking Agent A. When Agent A completes its operations and terminates, the TAREQ would terminate the session for Agent A, start a session for Agent B, and invoke Agent B.

Agent Notification Message

The TAREQ formats the agent notification message in accordance with the following data definition:

```
DEF agent-link.
  02 session-id.
  03 dummy PIC X(18).
  02 sender-info.
  03 sender-name PIC X(120).
  03 sender-applic-id PIC 9(4) COMP.
  02 recipient-info.
  03 recip-name PIC X(120).
  02 package-info.
  03 package-id.
  04 dummy PIC X(12).
  03 agent-selector PIC 9(4) COMP.
  03 package-flags.
    04 certified TYPE BOOLEAN.
    04 byte REDEFINES CERTIFIED PIC X.
    04 reserved-1 TYPE BOOLEAN VALUE "N".
    04 reserved-2 TYPE BOOLEAN VALUE "N".
    04 reserved-3 TYPE BOOLEAN VALUE "N".
    04 reserved-4 TYPE BOOLEAN VALUE "N".
    04 reserved-5 TYPE BOOLEAN VALUE "N".
    04 reserved-6 TYPE BOOLEAN VALUE "N".
    04 reserved-7 TYPE BOOLEAN VALUE "N".
  03 subject-string TYPE CHARACTER 140.
  02 depot-info.
  03 agent-data TYPE CHARACTER 80.
END.
```
In this definition, fields have the following meanings:

- **SESSION-ID** is the ID of the session required by the agent if the depot profile indicates that a session should be started on behalf of the agent. If no session is required, this field contains binary zeros.

- **SENDER-NAME** is the name of the correspondent who sent the package.

- **SENDER-APPLIC-ID** is the numeric application ID that identifies the client who transmitted the package with TISERV. This value is matched against the agent selection range established by APPLIC-ID-LOW and APPLIC-ID-HIGH in the depot profile records.

- **RECIP-NAME** is the name of the recipient to whom the package is sent.

- **PACKAGE-ID** is the ID of the header item for the package.

- **AGENT-SELECTOR** is the numeric value to be matched against the agent selection range established by AGENT-SEL-LOW and AGENT-SEL-HIGH in the depot profile records. The value used by the Tandem T/MAIL client is 0 (for DEFAULT-PKG).

- **PACKAGE-FLAGS** specifies the delivery control flags described as DELIV-CONTROL-FLAGS in the ALTER-ITEM-DESCR UOW in Section 5.

- **SUBJECT-STRING** is a data string that typically describes the contents of the package. (The MAIL client generates this string as its SUBJECT field.) If more than 140 characters are entered in this field, the additional characters are truncated.

- **AGENT-DATA** is an array containing data to be passed to the agent for use by the agent. This data is obtained from the Profile file. Your application establishes the data in that file through the ALTER-AGENT-SELECT UOW.

Upon receiving the notification message, the agent must respond by sending a reply to the TAREQ in accordance with the following format:

```plaintext
DEF agent-link-reply.
02 error-info.
   03 error-return PIC S9999 COMP.
   03 error-msg TYPE CHARACTER 80.
END.
```
Interfacing with TRANSFER

In this definition, fields have the following meanings:

- **ERROR-RETURN** notifies the TAREQ of any action it should take at this point. This field is required and must contain one of the following values:

<table>
<thead>
<tr>
<th>Interpretation</th>
<th>Value</th>
<th>TAREQ Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>GO-TO-NEXT-AGENT</td>
<td>0</td>
<td>Go to the next agent in the sequence. If TAREQ started the session for this agent, TAREQ ends the session.</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>Go to the next agent in the sequence. If TAREQ started the session for this agent, TAREQ does not end the session.</td>
</tr>
<tr>
<td>DON'T-GO-TO-NEXT-AGENT</td>
<td>1</td>
<td>Discontinue the search. If TAREQ started the session for this agent, TAREQ ends the session.</td>
</tr>
<tr>
<td></td>
<td>101</td>
<td>Discontinue the search. If TAREQ started the session for this agent, TAREQ does not end the session.</td>
</tr>
<tr>
<td>AGENT-ERROR</td>
<td>2</td>
<td>Log the error detected by the agent and invoke the next agent. If TAREQ started the session for this agent, TAREQ ends the session.</td>
</tr>
<tr>
<td></td>
<td>102</td>
<td>Log the error detected by the agent. If TAREQ started the session for this agent, TAREQ does not end the session.</td>
</tr>
<tr>
<td>AGENT-RESTART</td>
<td>3</td>
<td>Log the error detected by the agent and restart the current transaction.</td>
</tr>
</tbody>
</table>
WARNING

To avoid degrading system performance, AGENT-RESTART (3) should be returned only for a transient problem, such as a file lock. Returning the AGENT-RESTART for a problem such as record-not-found is not useful because the problem will not go away. AGENT-RESTART might cause previously successful deliveries to be backed out and then performed again.

For additional ramifications of TMF restart, refer to TAREQ Interface in Section 6.

- ERROR-MSG contains ASCII text that describes any error encountered. This field is optional. The text is written to the scheduler process log file if logging to this file is enabled.

The TAREQ calls a SCREEN COBOL agent with a CALL statement written in the following format:

CALL agent-name USING AGENT-LINK, AGENT-LINK-REPLY.

The LINKAGE SECTION of the user-supplied SCREEN COBOL program unit must correspond to this CALL statement.

The TAREQ sends to a server class agent with a SEND statement written in the following format:

SEND AGENT-LINK
TO agent-name
REPLY CODE 0,1,2 YIELDS AGENT-LINK-REPLY.

The user-supplied server must structure its definitions to agree with this SEND statement.

TAREQ Event Packages

When certain events occur, the TAREQ generates TAREQ event packages in accordance with certain predefined formats. In these packages, the application ID is always TAREQ (100). The agent selector value indicates the event that triggered the package. The events are listed in Table 4-3.
Table 4-3. TAREQ Events

<table>
<thead>
<tr>
<th>Agent-Selector Value</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TRANSFER SYSTEM ERROR</td>
</tr>
<tr>
<td>2</td>
<td>RECIPIENT HAS EXAMINED CERTIFIED PACKAGE</td>
</tr>
<tr>
<td>3</td>
<td>PACKAGE HAS INVALID RECIPIENT</td>
</tr>
<tr>
<td>4</td>
<td>PACKAGE HAS DISTRIBUTION LIST WITH INVALID RECIPIENT</td>
</tr>
<tr>
<td>5</td>
<td>PACKAGE HAS RECIPIENT WITH INVALID AGENT</td>
</tr>
<tr>
<td>6</td>
<td>PACKAGE COULD NOT BE DELIVERED TO RECIPIENT IN TIME</td>
</tr>
<tr>
<td>7</td>
<td>RECIPIENT DID NOT EXAMINE PACKAGE BEFORE IT EXPIRED</td>
</tr>
<tr>
<td>8</td>
<td>RECIPIENT HAS ALREADY EXAMINED CANCELED PACKAGE</td>
</tr>
<tr>
<td>9</td>
<td>SENDER CANCELED PACKAGE AFTER YOU EXAMINED IT</td>
</tr>
<tr>
<td>10</td>
<td>PACKAGE CANNOT BE TRANSPORTED TO RECIPIENT'S SYSTEM</td>
</tr>
<tr>
<td>11</td>
<td>AGENT IS MISBEHAVING</td>
</tr>
<tr>
<td>12</td>
<td>AGENT HAS LOGGED AN ERROR</td>
</tr>
</tbody>
</table>

The overall package format is illustrated in Figure 4-2. The format of the text item within this package depends on the agent selector value (event), as indicated in Figure 4-3.

For those TAREQ events that reflect an error, the TAREQ might not generate an event package, depending on the ERR-PKG-SUPPRESS-FLAGS field of the sender-submitted package. These flags are set by the sender.
Figure 4-2. Overall TAREQ Event Package Format
Interfacing with TRANSFER

Each format has four 120-byte text records

**FORMAT 1:**
AGENT-SELECTOR = Event
(1, 2, 3, 6, 7, 8 or 10)
(To sender)

<table>
<thead>
<tr>
<th>REC-SEQ-NUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>(blank)</td>
</tr>
</tbody>
</table>

| 2 |
| Recipient: |

| 3 |
| <recipient name> |

| 4 |
| (blank) |

**FORMAT 2:**
AGENT-SELECTOR = Event
(9)
(To recipient)

<table>
<thead>
<tr>
<th>REC-SEQ-NUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>(blank)</td>
</tr>
</tbody>
</table>

| 2 |
| Sender:    |

| 3 |
| <sender name> |

| 4 |
| (blank) |

**FORMAT 3:**
AGENT-SELECTOR = Event
(4)
(To sender)

<table>
<thead>
<tr>
<th>REC-SEQ-NUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>Distribution List:</td>
</tr>
</tbody>
</table>

| 2 |
| <distribution list name> |

| 3 |
| Recipient List: |

| 4 |
| <recipient list name> |

**FORMAT 4:**
AGENT-SELECTOR = Event
(5, 11, 12)
(To recipient)

<table>
<thead>
<tr>
<th>REC-SEQ-NUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>Agent:</td>
</tr>
</tbody>
</table>

| 2 |
| <agent name> |

| 3 |
| Sender: |

| 4 |
| <sender name> |

1 Record is intended only for output to users and need not be read by your application processes.

Figure 4-3. Text Portion Formats
Delivery Errors

TAREQs write delivery error indications to the recipient records involved. These error indications have the data type UOW-RETN-CODE; they can reflect almost any value returned in the RETN-CODE field by TISERV UOWs or they can reflect special delivery errors detected by TAREQs, whose values range from 4600 through 4699. A code of zero indicates that the TAREQ detected no error.

The following errors are detected by TAREQs; error explanations appear in Appendix A:

<table>
<thead>
<tr>
<th>Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4600</td>
<td>W-NOT-PROCESSED-HERE</td>
</tr>
<tr>
<td>4601</td>
<td>W-XPORTED</td>
</tr>
<tr>
<td>4602</td>
<td>E-XPORT-FAILED</td>
</tr>
<tr>
<td>4603</td>
<td>E-TOO-LATE-TO-XPORT</td>
</tr>
<tr>
<td>4604</td>
<td>W-FULLY-EXPANDED-DLIST</td>
</tr>
<tr>
<td>4605</td>
<td>E-CANCELED-UNEXAMINED</td>
</tr>
<tr>
<td>4607</td>
<td>E-EXPIRED-UNEXAMINED</td>
</tr>
<tr>
<td>4608</td>
<td>E-TOO-LATE-TO-DELIV</td>
</tr>
<tr>
<td>4609</td>
<td>E-INCONSISTENT-RECIP</td>
</tr>
</tbody>
</table>

Precautions about Using Agents

The use of agents can provide many advantages to your TRANSFER application. In designing your application, however, you should consider the following:

1. The amount of time that TRANSFER waits for agents can severely impact the TAREQ ability to perform other tasks. The TAREQ must wait for the agent to complete its activities, and thus cannot provide other services in the meantime. When the agent is a server rather than a SCREEN COBOL requester, you can specify time limits that the TAREQ must wait on the agent server class before aborting the transaction.

2. Multiple agents for a single recipient all operate within the same TMF transaction.

3. An agent required by a delivery is invoked only if that delivery is successful.
This section describes the available TISERV units-of-work (UOWs). The description of each UOW includes the following elements:

- the DDL format and syntax of the request UOW and its corresponding response UOW; the letters rsp at the beginning of each response UOW stand for response
- the content, function, and constraints of the individual fields transmitted within, or returned by, the UOW
- the operations performed by TISERV in response to the UOW.

Information transmitted to TISERV by a UOW is moved into the individual fields of the UOW by your program. Information returned in a UOW response is entered in the individual fields of the response by TISERV.

FILLER fields appear in the DDL format for certain UOWs. These fields provide for the alignment of fields on word boundaries in memory or allow space for the expansion of data in a field.

You can modify array limits in standard DDL definitions that contain variable length arrays. You can have multiple definitions with different array limits for the same UOW as long as the symbolic names for the UOW are unique.

It is recommended that IPCs be issued within the framework of TMF transactions. Some UOWs, however, do not change the TRANSFER data base and do not require a TMF transaction to be in effect. You do not need to start a TMF transaction before issuing the IPC if your IPC consists solely of one or more of the following UOWs:

- GET-AGENT-SELECT
- GET-CONFIG-NAME
- GET-FOLDER-ORDER
- GET-ITEM-DESCR
- GET-ITEM-REC
- GET-NEXT-SESSION
- GET-PROFILE-ELEM
- GET-RECIP-REC
- READ-PROFILE-REC-A02
- READ-NEXT-MEMBER
- READ-NEXT-NAME
- READ-PROF-REC
- SCAN-FOLDER
- SCAN-FOLDER-B00
- SCAN-FOLDER-BY-KEY
- WHERE-SAVED
UOW Descriptions

UOW HEADER AND RETURN CODE

Each UOW transmitted to TISERV begins with a UOW header. The DDL definition for this header is:

```
DEF uow-hdr.
  02 self-ident PIC AA VALUE "UW".
  02 uow-code TYPE BINARY 16 UNSIGNED.
END.
```

Individual fields in the UOW header contain the following information:

- SELF-IDENT always contains the characters UW to identify the header as a UOW header.

- UOW-CODE is a code value that identifies the specific UOW request being made. For example, an ADD-ITEM-REC UOW that adds a record to an item has a UOW-CODE field that contains the value 104.

Each response UOW returned from TISERV begins with the same header as the request, followed by two fields dealing with return codes. The DDL definition for this response is:

```
DEF uow-hdr.
  02 self-ident PIC AA VALUE "UW".
  02 uow-code
  DEF uow-retn-code
  DEF retn-code-detail
END.
```

```
TYPE BINARY 16 UNSIGNED.
TYPE BINARY 16.
TYPE BINARY 16.
```

Individual fields in the response UOW header contain the following additional information:

- UOW-RETN-CODE is the return code.
  - If no errors were encountered, this field is set to 0.
  - If an error occurred, this field contains a positive value ranging from 4000 through 5999; these values indicate the UOW was not processed.
  - If a warning was indicated, the field contains a negative value ranging from -5999 through -4000; these values indicate the UOW was successfully processed.

In the RETN-CODE field for each UOW, entries that begin with E denote errors returned to your process by TISERV, and entries that begin with W denote warnings. All possible entries are listed for each UOW. These errors are listed and described in Appendix A.
• RETN-CODE-DETAIL is a code that primarily identifies an error detected by a subsystem other than TRANSFER, such as the GUARDIAN operating system or the EXPAND network software, and for which TRANSFER provides no standard handling. These errors are discussed further in Appendix A.

All request and response UOWs must be aligned on word boundaries.

SOFTWARE PROVIDED WITH THE TRANSFER DELIVERY SYSTEM

The TRANSFER software release provides three files for interfacing with TRANSFER: GCOB, GLNK, and GDDL. These files contain source code for commonly used TRANSFER elements whose field and structure definitions appear in this manual. Typical TRANSFER elements are the interprocess communication (IPC) header that initiates a request, and the unit-of-work (UOW) definitions that describe operations to be performed.

This code can be copied into a SCREEN COBOL or COBOL source program by coding the record level and then using the COPY statement

COPY copy-text OF "filename"

where

    copy-text is the unique name for the definition in the named file. You determine the correct copy-text name by searching the files.

GCOB - This file contains COBOL source code for Working-Storage Section definitions.

GLNK - This file contains the same code as the GCOB file, but without the INITIAL-VALUE clauses. The GLNK file is used for Linkage Section definitions.

GCOB and GLNK have information in addition to the information produced by DDL when generating COBOL. The primary additions are prefixes for all field names, and many level 88 declarations for fields.

GDDL - This file contains DDL code that is used to create definitions for use when programming in a language other than SCREEN COBOL or COBOL.
UOW Descriptions

GDDL code does not exactly match the DDL syntax shown in this manual, but it should be immediately obvious how to interpret any differences.

UOW DEFINITIONS

The remainder of this section presents the available UOWs in alphabetic order. Each definition begins with the DDL format for the UOW request and corresponding response, followed by a description of the fields and the operations performed.

For programs that are written in languages other than COBOL and SCREEN COBOL, fields defined as PIC 9(4) COMP are limited to values in the range of 0 through 9999. In TAL, for example, these fields would have a type of INT, but entering a value outside the range would return an error.

Every UOW can return the following errors:

4010 E-BAD-TRANSACTION
4990 E-IO-TIMEOUT

Most UOWs can return some of the following errors:

4902 E-ERR-PROFILE-FILE
4904 E-ERR-SESSION-FILE
4906 E-ERR-ITEMDESC-FILE
4908 E-ERR-ITEMDATA-FILE

4910 E-ERR-RECIP-FILE
4912 E-ERR-FOLDER-FILE
4914 E-ERR-DLIST-FILE
4922 E-ERR-INV-FOLDER-FILE

In all of these cases you will receive a RQST-ERR, and the IPC-RETN-CODE and IPC-RETN-CODE-DETAIL will contain copies of the UOW RETN-CODE and RETN-CODE-DETAIL.
ACK-RECEIPT (UOW Code 131)

ACK-RECEIPT acknowledges the receipt of a package. This operation is recommended whenever a client retrieves a package from the INBOX folder; for example, whenever the client displays the contents of a package on behalf of a correspondent.

```plaintext
DEF ack-receipt-uow.
  02 hdr.
   03 self-ident PIC AA VALUE "UW".
   03 uow-code TYPE BINARY 16 UNSIGNED VALUE 131.
  02 item-id.
   03 dummy PIC X(12).
END.

DEF ack-receipt-rsp.
  02 hdr.
   03 self-ident PIC AA VALUE "UW".
   03 uow-code TYPE BINARY 16 UNSIGNED VALUE 131.
  02 retn-code TYPE BINARY 16.
  02 retn-code-detail TYPE BINARY 16.
END.
```

ACK-RECEIPT FIELDS. The fields defined in this UOW are:

- **HDR** is the UOW header. The UOW-CODE value is 131.

- **ITEM-ID** is the item ID of the package header for the package that was received.

- **RETN-CODE** is the return code. TISERV returns a code in this field to indicate one of the following entries:

  - 0 OK
  - 4010 E-BAD-_TRANSACTION
  - 4035 E-ITEM-NOT-FOUND
  - 4042 E-ITEM-NOT-PKG-HDR
  - 4045 E-TSCHED-UNAVAIL
  - 4080 E-PKG-NOT-RECEIVED
  - 4084 E-PKG-NOT-SUBMITTED
  - 4094 E-PKG-CANCELED
  - 4095 E-PKG-EXPIRED

- **RETN-CODE-DETAIL** is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.
UOW Descriptions
ACK-RECEIPT

ACK-RECEIPT OPERATION. ACK-RECEIPT sets the EXAMINED flag in the correspondent's recipient record of the package identified by ITEM-ID. This has the following effect:

- If the package was flagged by the sender for certification and the ACK-RECEIPT UOW is issued against this package for the first time by this recipient, TRANSFER transmits an acknowledgement package to the sender. Refer to the ALTER-ITEM-DESCR UOW for package certification details.

- When the expiration date is reached, TRANSFER checks the EXAMINED flag:
  - If this flag is set, TRANSFER performs no action.
  - If this flag is not set, TRANSFER removes the package from the INBOX if the package is still there, and notifies the sender that the package was not examined by the recipient.
ADD-ITEM-REC (UOW Code 104)

ADD-ITEM-REC adds a data record to an item.

---

```
DEF add-item-rec-uow.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 104.
  02 item-key.
    03 item-id.
      04 dummy PIC X(12).
    03 rec-type PIC 9(4) COMP.
    03 rec-seq-num PIC 9(4) COMP.
  02 data-byte-count TYPE BINARY 16 UNSIGNED.
  02 client-data.
    03 element PIC X OCCURS 0 TO 2000 TIMES DEPENDING ON data-byte-count.
END.
```

---

```
DEF add-item-rec-rsp.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 104.
  02 retn-code TYPE BINARY 16.
  02 retn-code-detail TYPE BINARY 16.
  02 rec-seq-num PIC 9(4) COMP.
END.
```

---

**ADD-ITEM-REC FIELDS.** The fields defined in this UOW are:

- **HDR** is the UOW header. The UOW-CODE value is 104.
- **ITEM-KEY** identifies the item to which the record will be added.
- **ITEM-ID** is the item ID of the item to which the record will be added.
- **REC-TYPE** is the type assigned to the record by your application.
  - Values 100 through 999 - reserved for use by Tandem
  - Values 1000 through 9999 - available for customer use
UOW Descriptions
ADD-ITEM-REC

REC-TYPE is not interpreted by TRANSFER, but is stored with the item record and is returned by the GET-ITEM-REC UOW. Records are stored first in increasing order by REC-TYPE, and within each REC-TYPE, in increasing order by REC-SEQ-NUM.

• REC-SEQ-NUM is the sequence number to be assigned to the record. The number can range from 1 through 9999; the value 0 is invalid for this field.

To assign the next highest unused sequence number within the record type specified, set REC-SEQ-NUM to -1. In the response, the sequence number that TISERV assigned to the record is returned. If the item contains no records, TISERV assigns the sequence number 1 to the new record.

• DATA-BYTE-COUNT is the length, in bytes, of the record to be added. This must be a value ranging from 0 through 2000.

• CLIENT-DATA is the data record to be added. The length of this data must be consistent with DATA-BYTE-COUNT.

NOTE

UOWs must start on word boundaries. If DATA-BYTE-COUNT contains an odd value and other UOWs follow this one in the request, you must append a one-byte FILLER to CLIENT-DATA.

• RETN-CODE is the return code. TISERV returns a code in this field to indicate one of the following entries:

   0    OK
 4010  E-BAD-TRANSACTION
 4035  E-ITEM-NOT-FOUND
 4041  E-ITEM-UNALTERABLE
 4046  E-INVALID-REC-TYPE
 4049  E-REC-ALREADY-EXISTS
 4058  E-INVALID-REC-SEQ-NUM
 4085  E-DATA TOO LONG

• RETN-CODE-DETAIL is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

ADD-ITEM-REC OPERATION. ADD-ITEM-REC adds the data record identified by DATA-BYTE-COUNT and CLIENT-DATA to the item identified by ITEM-KEY. TISERV assigns the sequence number indicated by REC-SEQ-NUM to this record. If you set REC-SEQ-NUM to -1, TISERV assigns the next highest unused sequence number within the record type specified.
Adding a record to an item is illustrated in Figure 5-1.

Figure 5-1. Adding a Record to an Item
ADD-MEMBER

ADD-MEMBER (UOW Code 220)

ADD-MEMBER adds one or more members to a distribution list. These members can be correspondents, distribution lists, or both.

```
DEF add-member-uow.
 02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 220.
    02 corr-name PIC X(80) VALUE SPACES.
    02 dlist-name PIC X(80).
    02 num-wanted TYPE BINARY 16 UNSIGNED VALUE 1.
    02 member-name PIC X(120)
        OCCURS 0 TO 5 TIMES
        DEPENDING ON num-wanted.
END.
```

```
DEF add-member-rsp.
 02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 220.
    02 retn-code TYPE BINARY 16.
    02 retn-code-detail TYPE BINARY 16.
    02 corr-name PIC X(80).
    02 dlist-name PIC X(80).
    02 num-returned TYPE BINARY 16 UNSIGNED.
    02 mbr-retn-code OCCURS 0 TO 5 TIMES
        DEPENDING ON num-returned
        TYPE BINARY 16.
END.
```

ADD-MEMBER FIELDS. The fields defined in this UOW are:

- HDR is the UOW header. The UOW-CODE value is 220.

- CORR-NAME is the name of the correspondent represented by the calling process. Only system administrators can enter correspondent names other than their own in this field; correspondents without system administrator privileges must enter their own names or leave the field blank. On input, you can use a partially qualified name, including wildcard characters. In the response, the fully qualified name is returned.
UOW Descriptions
ADD-MEMBER

- **DLIST-NAME** is the name of the distribution list to which the new member is added. This can be a partially qualified name, including wildcard characters. In the response, the fully qualified name is returned.

- **NUM-WANTED** is the number of members to be added to the distribution list. You can specify any number of members, restricted only by the length of the entire IPC that contains this UOW. The maximum IPC length is defined during TRANSFER system configuration.

- **MEMBER-NAME** is the name of the correspondent or distribution list to be added to the list identified by DLIST-NAME. This can be a partially qualified name provided local name resolution is not deferred in the depot profile. The name can include wildcard characters and a suffix. In the OCCURS DEPENDING ON clause, the value 5 is an arbitrary value suitable for most applications; you can reset it to any other value. You can have as many member names as specified by NUM-WANTED, which must be consistent with your DDL definition.

- **RETN-CODE** is the return code. TISERV returns a code in this field to indicate one of the following entries.

  To indicate successful addition of all members:

  0 OK

  To indicate problems with the correspondent name:

  5600 E-CORR-NSRV-ERR
  5601 E-CORR-NOT-FOUND
  5602 E-CORR-BAD-NAME
  5604 E-CORR-NO-SUCH-_NODE
  5606 E-CORR-NSRV-NOT-FOUND

  To indicate problems with the distribution list name:

  5625 E-DLIST-NSRV-ERR
  5626 E-DLIST-NOT-FOUND
  5627 E-DLIST-BAD-NAME
  5629 E-DLIST-NO-SUCH-NODE
  5631 E-DLIST-NSRV-NOT-FOUND

  To indicate at least one of the members was not added:

  4227 W-ERR-ON-MEMBER

  To indicate other problems:

  4010 E-BAD-TRANSACTION
  4093 E-SECURITY-VIOLATION
  4201 E-CONTEXT-ERR
UOW Descriptions
ADD-MEMBER

• RETN-CODE-DETAIL is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

• NUM-RETURNED is the number of members added to the list. There will be a member return code for each member that you attempted to add.

• MBR-RETURN-CODE is a code that indicates the status of each member the UOW attempted to add. The message OK indicates that the member was successfully added.

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OK</td>
</tr>
<tr>
<td>4067</td>
<td>W-REMOTE-NAME-ACCEPTED</td>
</tr>
<tr>
<td>4069</td>
<td>W-NODE-NAME-UNKNOWN</td>
</tr>
<tr>
<td>5650</td>
<td>E-MBR-NSRV-ERR</td>
</tr>
<tr>
<td>5651</td>
<td>E-MBR-NOT-FOUND</td>
</tr>
<tr>
<td>5652</td>
<td>E-MBR-BAD-NAME</td>
</tr>
<tr>
<td>5654</td>
<td>E-MBR-NO-SUCH-NODE</td>
</tr>
<tr>
<td>5656</td>
<td>E-MBR-NSRV-NOT-FOUND</td>
</tr>
<tr>
<td>5657</td>
<td>E-MBR-NSRV-DOWN</td>
</tr>
<tr>
<td>5661</td>
<td>E-MBR-NET-DOWN</td>
</tr>
<tr>
<td>5662</td>
<td>E-MBR-ALREADY-EXISTS</td>
</tr>
<tr>
<td>5673</td>
<td>E-MBR-AMBIGUOUS-NAME</td>
</tr>
<tr>
<td>5674</td>
<td>E-MBR-BAD-SUFFIX</td>
</tr>
</tbody>
</table>

ADD-MEMBER OPERATION. ADD-MEMBER adds the new members identified by MEMBER-NAME to the distribution list identified by DLIST-NAME. You can specify any number of new members even though the DDL format shows only 5, restricted only by the limit specified in the OCCURS clause and the length of the entire IPC that contains this UOW. The maximum IPC length is defined during TRANSFER system configuration.

The names of the new distribution list members are fully resolved as specified by flags in the depot profile. If remote name resolution is specified in the profile and you add many names to one or more remote systems, the resulting network traffic might slow your application appreciably.

If the depot profile specifies that local name resolution is deferred, the new member names must be fully qualified and syntactically correct.

If the depot profile specifies that remote resolution is deferred, a new member name that specifies a remote node will be added with a warning error:

- W-REMOTE-NAME-ACCEPTED if the node name is recognized
- W-NODE-NAME-UNKNOWN if the node name is not currently known to EXPAND

A member name can include a suffix. If the name is resolved to be that of another distribution list or that of a correspondent whose depot profile indicates the depot currently does not accept suffixes, the suffix is discarded without notice.
ADD-RECIP (UOW Code 114)
ADD-RECIP adds a recipient to a package recipient list.

```
DEF add-recipient-uow.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 114.
  02 item-id.
    03 dummy PIC X(12).
    02 recip-name PIC X(120).
    02 recip-type PIC 9(4) COMP.
  02 options.
    03 use-depot-resol-flags TYPE BOOLEAN.
    03 defer-local-resolution TYPE BOOLEAN.
    03 defer-remote-resolution TYPE BOOLEAN.
    03 derived-from-dlist TYPE BOOLEAN.
    03 reserved-4 TYPE BOOLEAN VALUE "N".
    03 reserved-5 TYPE BOOLEAN VALUE "N".
    03 reserved-6 TYPE BOOLEAN VALUE "N".
    03 reserved-7 TYPE BOOLEAN VALUE "N".
END.
```

```
DEF add-recipient-rsp.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 114.
  02 retn-code TYPE BINARY 16.
  02 retn-code-detail TYPE BINARY 16.
  02 accepted-name PIC X(120).
END.
```

ADD-RECIP FIELDS. The fields defined in this UOW are:

- HDR is the UOW header. The UOW-CODE value is 114.
- ITEM-ID identifies the package whose recipient list is to be modified. This is the item ID of the header for that package.
ADD-RECIP

- RECIP-NAME is the name of the recipient to be added to the recipient list. The name can identify either a correspondent or a distribution list. This can be a partially qualified name provided local and remote name resolution is not deferred in the depot profile. The name can include wildcard characters and a suffix.

If the name is resolved to be that of a distribution list or that of a correspondent whose depot profile indicates the depot currently does not accept suffixes, the suffix is discarded without notice.

- RECIP-TYPE is the recipient type by which your application categorizes the recipients of packages, as described in Appendix B. This field is not interpreted by TRANSFER, but is returned by the GET-RECIP-REC UOW.

- OPTIONS includes the following fields:
  
  USE-DEPOT-RESOL-FLAGS

  Y = TRANSFER resolves the recipient name by using the depot service flags pertaining to name resolution and defined at the depot with the ALTER-PROFILE-ELEMENT UOW, and ignores the DEFER-LOCAL-RESOLUTION and DEFER-REMOTE-RESOLUTION fields.

  N = TRANSFER ignores those depot service flags and uses the DEFER-LOCAL-RESOLUTION and DEFER-REMOTE-RESOLUTION fields as name resolving criteria.

  DEFER-LOCAL-RESOLUTION

  Y = TRANSFER defers the resolution of locally defined names.

  N = TRANSFER immediately resolves these names.

  DEFER-REMOTE-RESOLUTION

  Y = TRANSFER defers the resolution of remotely defined names.

  N = TRANSFER immediately resolves these names.

  DERIVED-FROM-DLIST is used only by TAREQs; this field must be set to N.

  RESERVED-4 through RESERVED-7 are reserved for use by Tandem; these fields must be set to N.
ADD-RECIP

UOW Descriptions

- **RETN-CODE** is the return code. TISERV returns a code in this field to indicate one of the following entries.

  To indicate successful addition of the recipient:
  
  0 OK

  To indicate problems with the recipient name:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5750</td>
<td>E-RECIP-NSRV-ERR</td>
</tr>
<tr>
<td>5751</td>
<td>E-RECIP-NOT-FOUND</td>
</tr>
<tr>
<td>5752</td>
<td>E-RECIP-BAD-NAME</td>
</tr>
<tr>
<td>5754</td>
<td>E-RECIP-NO-SUCH-NODE</td>
</tr>
<tr>
<td>5755</td>
<td>E-RECIP-SECURITY</td>
</tr>
<tr>
<td>5756</td>
<td>E-RECIP-NSRV-NOT-FOUND</td>
</tr>
<tr>
<td>5757</td>
<td>E-RECIP-NSRV-DOWN</td>
</tr>
<tr>
<td>5761</td>
<td>E-RECIP-NET-DOWN</td>
</tr>
<tr>
<td>5773</td>
<td>E-RECIP-AMBIGUOUS-NAME</td>
</tr>
<tr>
<td>5774</td>
<td>E-RECIP-BAD-SUFFIX</td>
</tr>
</tbody>
</table>

  To indicate other problems:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4010</td>
<td>E-BAD-TRANSACTION</td>
</tr>
<tr>
<td>4035</td>
<td>E-ITEM-NOT-FOUND</td>
</tr>
<tr>
<td>4041</td>
<td>E-ITEM-UNALTERABLE</td>
</tr>
<tr>
<td>4042</td>
<td>E-ITEM-NOT-PKG-HDR</td>
</tr>
<tr>
<td>4049</td>
<td>W-REC-ALREADY-EXISTS</td>
</tr>
<tr>
<td>4051</td>
<td>E-MUST-BE-YN</td>
</tr>
<tr>
<td>4052</td>
<td>E-RESERVED-MUST-BE-N</td>
</tr>
<tr>
<td>4065</td>
<td>E-INVALID-RECIP-TYPE</td>
</tr>
<tr>
<td>4067</td>
<td>W-REMOTE-NAME-ACCEPTED</td>
</tr>
<tr>
<td>4069</td>
<td>W-NODE-NAME-UNKNOWN</td>
</tr>
</tbody>
</table>

- **RETN-CODE-DETAIL** is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

- **ACCEPTED-NAME** is the recipient name returned by TRANSFER, resolved as fully as possible at this time.

**ADD-RECIP OPERATION.** ADD-RECIP adds the name identified by RECIP-NAME to the recipient list for the package identified by ITEM-ID.

You can request TRANSFER to defer resolution of local or remote recipient names with the DEFER-LOCAL-RESOLUTION or DEFER-REMOTE-RESOLUTION fields, respectively. If resolution deferral is requested for either a local or remote name, that name must be supplied as a fully qualified name in the RECIPIENT-NAME field; in this case, the name will be interactively checked for correct syntax only.

Distribution list names are always expanded asynchronously.
ALTER-AGENT-SELECT (UOW Code 208)

ALTER-AGENT-SELECT changes agent selection criteria for a depot. Typically, this UOW is issued after a GET-AGENT-SELECT UOW.

DEF alter-agent-select-uow.
  02 hdr.
     03 self-ident PIC AA VALUE "UW".
     03 uow-code TYPE BINARY 16 UNSIGNED VALUE 208.
     02 filler
     02 corr-name PIC X(80) VALUE SPACES.
     02 filler
     02 action
     02 agent-rec-num TYPE BINARY 16 UNSIGNED.
     02 agent-name TYPE CHARACTER 80.
     02 agent-type TYPE CHARACTER 1.
     02 agent-flag TYPE CHARACTER 1.
     02 applic-id-for-logon PIC 9(4) COMP.
     02 applic-id-low PIC 9(4) COMP.
     02 applic-id-high PIC 9(4) COMP.
     02 agent-sel-low PIC 9(4) COMP.
     02 agent-sel-high PIC 9(4) COMP.
     02 agent-data TYPE CHARACTER 80.
END.

DEF alter-agent-select-rsp.
  02 hdr.
     03 self-ident PIC AA VALUE "UW".
     03 uow-code TYPE BINARY 16 UNSIGNED VALUE 208.
     02 retn-code TYPE BINARY 16.
     02 retn-code-detail TYPE BINARY 16.
     02 agent-rec-num PIC X(80).
     02 corr-name
END.

ALTER-AGENT-SELECT FIELDS. The fields defined in this UOW are:

- HDR is the UOW header. The UOW-CODE value is 208.
UOW Descriptions
ALTER-AGENT-SELECT

• CORR-NAME is the name of the correspondent represented by the calling process. Only system administrators can enter correspondent names other than their own in this field; correspondents without system administrator privileges must enter their own names or leave the field blank. On input, you can use a partially qualified name, including wildcard characters. In the response, the fully qualified name is returned.

• ACTION specifies the type of updating to be performed, as follows:

<table>
<thead>
<tr>
<th>Entry</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>INSERT-AFTER-REC (I)</td>
<td>Insert the selection criteria defined by UOW fields AGENT-NAME through AGENT-DATA. Place this criteria in the agent list after the record identified by AGENT-REC-NUM. To insert the record at the beginning of the list, set AGENT-REC-NUM to 0.</td>
</tr>
<tr>
<td>REWRITE-REC (W)</td>
<td>Update the agent specified by AGENT-REC-NUM.</td>
</tr>
<tr>
<td>DELETE-REC (D)</td>
<td>Delete the agent specified by AGENT-REC-NUM.</td>
</tr>
</tbody>
</table>

• AGENT-REC-NUM is the position of the record that defines the agent in the agent list for the depot. To get the position number, you can issue a GET-AGENT-SELECT UOW and perform a read-after with AGENT-REC-NUM set to 0.

In the response, TISERV sets this field as follows:
- For insert (I action), the field is set to the record number where the selection criteria was inserted.
- For update (W action), the field is set to the same record number entered on input.
- For delete (D action), the field is set to 0 if no agent exists after the one just deleted; or the field is set to the number of the next remaining agent, which is a value equal to that provided in the delete request.

• AGENT-NAME is the name of the SCREEN COBOL program or server class that operates as the agent. You determine whether this name identifies a SCREEN COBOL program or a server class with the AGENT-TYPE field.
• AGENT-TYPE determines whether the program denoted by AGENT-NAME is a SCREEN COBOL program or a server class name.

   Y = SCREEN COBOL program

   N = server class name

• AGENT-FLAG determines whether the agent requires the associated TAREQ to begin a session on its behalf before the agent can run. If the agent issues UOWs to a TRANSFER process, a session must be started for the agent before the agent can execute.

   Y = initiate session

   N = do not initiate session

• APPLIC-ID-FOR-LOGON is the application ID to be used in the START-SESSION UOW when the AGENT-FLAG field is set to Y, meaning that a session will be started before invoking the agent.

• APPLIC-ID-LOW, APPLIC-ID-HIGH, AGENT-SEL-LOW, and AGENT-SEL-HIGH define numeric ranges used to select the agent. These fields can contain values ranging from 0 through 9999. Agent selection is as follows:

   1. TAREQ examines the list of agents for a depot and compares the APPLIC-ID submitted with the package against the range defined by APPLIC-ID-LOW and APPLIC-ID-HIGH in the agent selection criteria. If the submitted APPLIC-ID falls within this range, TAREQ continues to Step 2; otherwise, the agent is not selected for execution.

   In these fields, the values 100 through 999 are reserved for Tandem.

   2. If the APPLIC-ID submitted with the package falls within the range defined by APPLIC-ID-LOW and APPLIC-ID-HIGH, TAREQ compares the AGENT-SEL field submitted with the package against the range defined by AGENT-SEL-LOW and AGENT-SEL-HIGH. If the submitted AGENT-SEL falls within this range, TAREQ invokes the agent.

• AGENT-DATA is data that your application passes to the agent; its use is defined by the agent itself.
NOTE

The default agent configuration screen allows for viewing and modifying only the first 78 characters of the 80-character AGENT-DATA field. If you are providing an agent that requires 79 or 80 characters of agent data, you must also provide a SCREEN COBOL program for configuring such agents.

• RETN-CODE is the return code. TISERV returns a code in this field to indicate one of the following entries.

To indicate successful alteration of the agent selection criteria:

0  OK

To indicate problems with the correspondent name:

5600  E-CORR-NSRV-ERR
5601  E-CORR-NOT-FOUND
5602  E-CORR-BAD-NAME
5604  E-CORR-NO-SUCH-NODE
5606  E-CORR-NSRV-NOT-FOUND

5607  E-CORR-NSRV-DOWN
5611  E-CORR-NET-DOWN
5622  E-CORR-NOT-SAME-NODE
5623  E-CORR-AMBIGUOUS-NAME

To indicate other problems:

4010  E-BAD-TRANSACTION
4047  E-REC-NOT-FOUND
4054  E-INVALID-AGENT-SEL
4055  E-INVALID-APPLIC-1D
4058  E-INVALID-REC-SEQ-NUM
4093  E-SECURITY-VIOLATION
4201  E-CONTEXT-ERR
4214  E-INVALID-AGENT-NAME
4230  E-MUST-BE-IWD
4231  E-INVALID-AGENT-FLAG
4232  E-INVALID-AGENT-TYPE
4902  E-ERR-PROFILE-FILE

• RETN-CODE-DETAIL is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

ALTER-AGENT-SELECT OPERATION. ALTER-AGENT-SELECT creates, deletes, or updates the requested agent selection entry recorded in the depot profile file. The agent is identified by AGENT-REC-NUM. The selection entry is contained in fields AGENT-NAME through AGENT-DATA.

The list of agents for a depot is ordered by agent record number (AGENT-REC-NUM field). Existing agents can have AGENT-REC-NUM values in the range of 1 through 500.
The AGENT-REC-NUM values are numbered consecutively. Deleting an agent causes the AGENT-REC-NUM of all subsequent agents to decrease by one. Inserting an agent causes the AGENT-REC-NUM of all later agents to increase by one.

To insert a new agent after an existing agent, set the ACTION field to I and the AGENT-REC-NUM field to the record number of the existing agent.

To insert an agent at the beginning of the agent list, set the ACTION field to I and the AGENT-REC-NUM field to 0.

A write operation returns the AGENT-REC-NUM of the agent just written. Thus, an insert operation returns the record number of the agent just inserted; the number is one greater than the number you provided in the insert request. To add the next agent to the end of the list, you should use this record number in a subsequent ALTER-AGENT-SELECT UOW.

A delete operation returns either of the following:

- zero if no agent exists after the one just deleted
- the number of the next remaining agent, which is a value equal to that provided in the delete request.
ALTER-FOLDER-ORDER (UOW Code 234)

ALTER-FOLDER-ORDER modifies the ordering criteria for an existing folder. The folder must be empty.

```
DEF alter-folder-order-uow.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 234.
  02 corr-name PIC X(80) VALUE SPACES.
  02 folder-name PIC X(80).
  02 ordering-discipline TYPE CHARACTER 1.
    88 time-saved VALUE "T".
    88 creator-name VALUE "C".
    88 earliest-deliv-date VALUE "E".
    88 applic-defined VALUE "A".
  02 filler PIC X(1) VALUE SPACES.
  02 applic-order-type PIC 9(4) COMP.
  02 options.
    03 ascending-sequence TYPE BOOLEAN.
    03 allow-duplicates TYPE BOOLEAN.
    03 reserved-2 TYPE BOOLEAN VALUE "N".
    03 reserved-3 TYPE BOOLEAN VALUE "N".
    03 reserved-4 TYPE BOOLEAN VALUE "N".
    03 reserved-5 TYPE BOOLEAN VALUE "N".
    03 reserved-6 TYPE BOOLEAN VALUE "N".
    03 reserved-7 TYPE BOOLEAN VALUE "N".
END.
```

```
DEF alter-folder-order-rsp.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 234.
  02 retn-code TYPE BINARY 16.
  02 retn-code-detail TYPE BINARY 16.
  02 corr-name PIC X(80).
  02 folder-name PIC X(80).
END.
```

ALTER-FOLDER-ORDER FIELDS. The fields defined in this UOW are:

- HDR is the UOW header. The UOW-CODE value is 234.
UOW Descriptions
ALTER-FOLDER-ORDER

- CORR-NAME is the name of the correspondent represented by the calling process. Only system administrators can enter correspondent names other than their own in this field; correspondents without system administrator privileges must enter their own names or leave the field blank. On input, you can use a partially qualified name, including wild card characters. In the response, the fully qualified correspondent name is returned.

- FOLDER-NAME is the name of an existing folder that is empty.

Special folder names INBOX and WASTEBASKET can be entered provided the special folders are empty.

- These special folders cannot have APPLIC-DEFINED (A) ordering discipline. An attempt to use APPLIC-DEFINED ordering for these folders returns the error E-SPECIAL-FLD.

- These special folders must allow duplicates. An attempt to set the ALLOW-DUPLICATES option to N for these folders returns the error E-SPECIAL-FLD.

- ORDERING-DISCIPLINE specifies what information TISERV should use as the ordering key when saving items in the folder as follows:

<table>
<thead>
<tr>
<th>Entry</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIME-SAVED (T)</td>
<td>Save items in chronological order of the time a SAVE-ITEM UOW is executed for the items.</td>
</tr>
<tr>
<td></td>
<td>TRANSFER A01/A02 uses the TIME-SAVED ordering discipline exclusively. Folders created by either TRANSFER A01 or A02 are ordered in ascending sequence by time saved unless the ordering criteria is subsequently altered.</td>
</tr>
<tr>
<td>CREATOR-NAME (C)</td>
<td>Save items in alphabetic order by creator name.</td>
</tr>
<tr>
<td>EARLIEST-DELIV-DATE (E)</td>
<td>Save items that are package headers and are unalterable by earliest delivery date; save items that are not package headers or are package headers and are alterable by creation date.</td>
</tr>
</tbody>
</table>
APPLIC-DEFINED (A) Save items in order of the ordering key specified by the application in the SAVE-ITEM-BY-KEY UOW.

An application can specify the same ordering key for multiple items if duplicates are allowed; if duplicates are not allowed, the application is responsible for ensuring the uniqueness of the key.

If this field contains a character other than T, C, E, or A, TISERV returns the error E-INVALID-ORD-DISCIPLN.

- APPLIC-ORDER-TYPE is meaningful only if ORDERING-DISCIPLINE is APPLIC-DEFINED (A). If ORDERING-DISCIPLINE is not APPLIC-DEFINED, the field is ignored.

This field is not interpreted by TRANSFER, but is stored as part of the folder's ordering criteria and is returned by the GET-FOLDER-ORDER UOW. This enables a TRANSFER application to have several types of APPLIC-DEFINED ordering; the APPLIC-ORDER-TYPE can be used to distinguish between types.

The field can have a value from 0 through 9999; values 100 through 999 are reserved for Tandem. If a number outside this range is specified, TISERV returns the error E-INVALID-APP-ORD-TYPE.

- OPTIONS provides additional criteria for saving items. Two options are provided.

  ASCENDING-SEQUENCE determines whether items are saved within the folder in ascending or descending key order.

  Y = Save items in ascending key order.

  N = Save items in descending key order.

Variable length APPLIC-DEFINED keys that are used with descending key order will not collate in the correct order unless the application pads the key with enough bytes to make it a constant length for all entries in the folder.
UOW Descriptions
ALTER-FOLDER-ORDER

A folder ordered by CREATOR-NAME in descending sequence is an example of a field that is variable length and is automatically padded to be 120 bytes. For a folder ordered by CREATOR-NAME in ascending sequence, the amount of disc space used correlates directly to the length of creator names and the number of items saved in a folder. Considering creator names of a maximum of 20 characters, the cost of descending sequence over ascending sequence in terms of disc space is approximately 100 bytes per item saved in a folder ordered by CREATOR-NAME.

ALLOW-DUPLICATES determines whether or not multiple items with the same ordering key can be saved in a folder.

Y = Duplicate ordering keys are allowed.

N = Duplicate ordering keys are not allowed. This field has no effect if the ORDERING-DISCIPLINE is TIME-SAVED (T); TIME-SAVED ordering already ensures a unique key. This option is illegal for special folders INBOX and WASTEBASKET.

RESERVED-2 through RESERVED-7 are reserved for use by Tandem; these fields must be set to N.

• RETN-CODE is the return code. TISERV returns a code in this field to indicate one of the following entries.

To indicate successful ordering of the folder:

0 OK

To indicate problems with the correspondent name:

5600 E-CORR-NSVR-ERR 5607 E-CORR-NSRV-DOWN
5601 E-CORR-NOT-FOUND 5611 E-CORR-NET-DOWN
5602 E-CORR-BAD-NAME 5622 E-CORR-NOT-SAME-NODE
5604 E-CORR-NO SUCH NODE 5623 E-CORR-AMBIGUOUS-NAME
5606 E-CORR-NSRV-NOT-FOUND

To indicate problems with the folder name:

5675 E-FLD-NSRV-ERR 5682 E-FLD-NSRV-DOWN
5676 E-FLD-NOT-FOUND 5683 E-FLD-NO-PARENT
5677 E-FLD-BAD-NAME 5686 E-FLD-NET-DOWN
5679 E-FLD-NO-SUCH-NODE 5697 E-FLD-NOT-SAME-NODE
5681 E-FLD-NSRV-NOT-FOUND 5698 E-FLD-AMBIGUOUS-NAME
To indicate other problems:

4010 E-BAD-TRANSACTION  4265 E-INVALID-ORD-DISCIPLN
4051 E-MUST-BE-YN    4266 E-INVALID-APP-ORD-TYPE
4052 E-RESERVED-MUST-BE-N  4267 E-ITEMS-IN-FLD
4093 E-SECURITY-VIOLATION  4968 E-SPECIAL-FLD
4105 E-CONCURRENT-FLD-UPDATE  4902 E-ERR-PROFILE-FILE
4201 E-CONTEXT-ERR    4912 E-ERR-FOLDER-FILE

• RETN-CODE-DETAIL is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

ALTER-FOLDER-ORDER OPERATION.  ALTER-FOLDER-ORDER updates the ordering criteria for the specified folder. Field values in this UOW indicate the new ordering criteria.

Folder order affects the sequence in which items are saved and subsequently referenced. Each item saved in a folder has an associated ordering key. The ordering key can be information associated with the item, such as the name of the item's creator, or information provided by the TRANSFER application. Every folder is ordered by the value of the ordering key associated with each item stored in the folder.

SCAN-FOLDER, SCAN-FOLDER-B00, and SCAN-FOLDER-BY-KEY UOWs retrieve item IDs in ordering key sequence. The SCAN-FOLDER-BY-KEY UOW, however, is restricted to folders with an APPLIC-DEFINED ordering discipline.

A folder must be empty before the ordering criteria can be altered. If an ALTER-FOLDER-ORDER UOW is issued for a folder that contains items, TISERV returns the error E-ITEMS-IN-FLD.
ALTER-ITEM-DESCR (UOW Code 116)

ALTER-ITEM-DESCR updates fields in an item descriptor. The descriptor can apply to either a package or a non-package item. This UOW is used primarily to set a number of attributes for a package prior to submission of that package for delivery.

Typically, this UOW is issued after a GET-ITEM-DESCR UOW.

```
DEF alter-item-descr-uow.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 116.
  02 item-id.
    03 dummy PIC X(12).
  02 update-control.
    02 fields-to-alter.
      02 item-type TYPE BOOLEAN.
      02 earliest-deliv-date TYPE BOOLEAN.
      02 latest-deliv-date TYPE BOOLEAN.
      02 expiration-date TYPE BOOLEAN.
      02 priority TYPE BOOLEAN.
      02 agent-selector TYPE BOOLEAN.
      03 reserved-6 TYPE BOOLEAN VALUE "N".
      03 reserved-7 TYPE BOOLEAN VALUE "N".
      03 deliv-control-flags.
        04 certified TYPE BOOLEAN.
        04 byte REDEFINES CERTIFIED PIC X.
        04 reserved-1 TYPE BOOLEAN VALUE "N".
        04 reserved-2 TYPE BOOLEAN VALUE "N".
        04 reserved-3 TYPE BOOLEAN VALUE "N".
        04 reserved-4 TYPE BOOLEAN VALUE "N".
        04 reserved-5 TYPE BOOLEAN VALUE "N".
        04 reserved-6 TYPE BOOLEAN VALUE "N".
        04 reserved-7 TYPE BOOLEAN VALUE "N".
      03 err-pkg-suppress-flags.
        04 invalid-recip TYPE BOOLEAN.
        04 byte REDEFINES INVALID-RECIPI PIC X.
        04 invalid-dlist TYPE BOOLEAN.
        04 reserved-2 TYPE BOOLEAN VALUE "N".
        04 too-late-to-deliver TYPE BOOLEAN.
        04 expired-unexamined TYPE BOOLEAN.
        04 reserved-5 TYPE BOOLEAN VALUE "N".
        04 reserved-6 TYPE BOOLEAN VALUE "N".
        04 reserved-7 TYPE BOOLEAN VALUE "N".
```
02 alterable-fields.
  03 item-type PIC 9(4) COMP.
  03 rel-date-earliest TYPE BOOLEAN.
  03 rel-date-latest TYPE BOOLEAN.
  03 rel-date-expiration TYPE BOOLEAN.
  03 reserved-3 TYPE BOOLEAN VALUE "N".
  03 earliest-deliv-date.
    04 date-time.
      05 year PIC 9(4).
      05 month PIC 9(2).
      05 day-of-month PIC 9(2).
      05 hour PIC 9(2).
      05 minute PIC 9(2).
      05 second PIC 9(2).
  04 delta-time REDEFINES DATE-TIME.
    05 quantity PIC 9(4) COMP.
    05 units PIC A.
    05 filler PIC X.
  03 latest-deliv-date.
    04 date-time.
      05 year PIC 9(4).
      05 month PIC 9(2).
      05 day-of-month PIC 9(2).
      05 hour PIC 9(2).
      05 minute PIC 9(2).
      05 second PIC 9(2).
  04 delta-time REDEFINES DATE-TIME.
    05 quantity PIC 9(4) COMP.
    05 units PIC A.
    05 filler PIC X.
  03 expiration-date.
    04 date-time.
      05 year PIC 9(4).
      05 month PIC 9(2).
      05 day-of-month PIC 9(2).
      05 hour PIC 9(2).
      05 minute PIC 9(2).
      05 second PIC 9(2).
  04 delta-time REDEFINES DATE-TIME.
    05 quantity PIC 9(4) COMP.
    05 units PIC A.
    05 filler PIC X.
  03 priority PIC 9(3) COMP.
  03 agent-selector PIC 9(4) COMP.
  03 deliv-control-flags.
    04 certified TYPE BOOLEAN.
    04 byte REDEFINES CERTIFIED PIC X.
UOW Descriptions
ALTER-ITEM-DESCR

ALTER-ITEM-DESCR FIELDS. The fields defined in this UOW are:

- HDR is the UOW header. The UOW-CODE value is 116.
- ITEM-ID identifies the item whose descriptor record is to be changed.
- UPDATE-CONTROL is the counter used to control concurrent update operations.

Enter the value returned by the GET-ITEM-DESCR UOW. If the UPDATE-CONTROL field is greater than 0, it must match the update control value stored in the data base for the update to succeed. If you set the UPDATE-CONTROL field to -1, TRANSFER does not check this field against the value stored in the data base. UPDATE-CONTROL values of 0 or less than -1 are invalid.
• FIELDS-TO-ALTER lets you select the fields to be altered within the item descriptor.

  \textbf{Y} = \text{Select the field for alteration.}

  \textbf{N} = \text{Disable the field from alteration.}

The meaning of each item descriptor field that can be accessed by your process is discussed under ALTERABLE-FIELDS. All of these fields can be retrieved through the \textsc{GET-ITEM-DESCR} UOW.

RESERVED-6 and RESERVED-7 are reserved for use by Tandem; these fields must be set to \textbf{N}.

The alterable fields do not require system administrator capability for alteration; if the item is alterable, the fields can be modified by any correspondent who has access to the item.

• DELIV-CONTROL-FLAGS specifies whether or not the package is certified. Fields within DELIV-CONTROL-FLAGS are as follows:

  \textbf{CERTIFIED} specifies whether the package is certified; that is, whether a certification package is returned to the sender whenever a recipient acknowledges receipt of a package via the \textsc{ACK-RECEIPT} UOW.

  \textbf{Y} = \text{Flag the package for certification.}

  \textbf{N} = \text{Do not flag the package for certification.}

RESERVED-1 through RESERVED-7 are reserved for use by Tandem; these fields must be set to \textbf{N}.

• ERR-PKG-SUPPRESS-FLAGS lets you suppress the effects of specific asynchronous errors. Fields within ERR-PKG-SUPPRESS-FLAGS are as follows:

  The flags INVALID-RECIP through EXPIRED-UNEXAMINED denote asynchronous errors detected by TRANSFER. If one of these errors occurs and error reporting is not suppressed, TRANSFER transmits a package reporting the error to the sender.

  \textbf{Y} = \text{Suppress the error-reporting package.}

  \textbf{N} = \text{Do not suppress the error-reporting package.}

The fields RESERVED-2 and RESERVED-5 through RESERVED-7 are reserved for use by Tandem; these fields must be set to \textbf{N}. 
ALTER-ITEM-DESCR

- ALTERABLE-FIELDS lets you indicate what updating should be done to the fields for which Y was specified under FIELDS-TO-ALTER. These fields and their allowable entries are defined as follows:

ITEM-TYPE indicates the type of the item, as defined by the application.

REL-DATE-EARLIEST, REL-DATE-LATEST, and REL-DATE-EXPIRATION determine whether the entries you specify in the EARLIEST-DELIV-DATE, LATEST-DELIV-DATE, and EXPIRATION-DATE fields are relative to the dates with respect to particular delivery milestones, or absolute calendar dates.

Y = relative dates

N = absolute dates

RESERVED-3 is reserved for use by Tandem; this field must be set to N.

EARLIEST-DELIV-DATE is the earliest date at which the package can be delivered, specified either as an absolute calendar date or as a number of time units from submission time. An absolute or relative date is indicated by the REL-DATE-EARLIEST field.

EARLIEST-DELIV-DATE and LATEST-DELIV-DATE together define the delivery window within which TRANSFER must deliver the package, as illustrated in Figure 5-2. If TRANSFER is unable to deliver the package to one or more recipients during the delivery window, it returns the package to the sender with an explanation and discontinues further delivery attempts. Under no circumstances is a package delivered sooner than its EARLIEST-DELIV-DATE. You can regard the EARLIEST-DELIV-DATE as the effective submission date of the package—in other words, as the postmark.

In DELTA-TIME, UNITS is the unit of time denoted in QUANTITY. UNITS can be set to D (for days), H (for hours), and M (for minutes). To specify one day, for example, set QUANTITY to 1 and UNITS to D.

LATEST-DELIV-DATE is the latest date at which TRANSFER can deliver the package, specified either as an absolute calendar date or as a number of time units from EARLIEST-DELIVERY-DATE. An absolute or relative date is indicated by the REL-DATE-LATEST field.
EXPIRATION-DATE is the date on which the package expires. On this date, the package is automatically removed from the INBOX of any recipient who has not already acknowledged its receipt through the ACK-RECEIPT UOW. Packages that either are acknowledged or are in other folders are not affected by expiration. Any copy of the package kept by the sender must be explicitly discarded by the sender. A relative date of value 0 M means no expiration.

If a package expires and the recipient has not acknowledged its receipt, TRANSFER removes the package from the INBOX and notifies the sender. Packages that are acknowledged by the recipient are never removed from the INBOX by TRANSFER; they must be explicitly removed by the recipient.

By initial system default, TRANSFER packages do not expire. You can specify that a given package should eventually expire by including a nonzero absolute or relative expiration date/time in the package header. You can change the default for your depot by specifying a nonzero relative expiration time in your correspondent profile. This value must be within the maximum lifespan as indicated by a TRANSFER system control parameter; each package submitted with a nonzero expiration field will have that field increased, as necessary, so that it exceeds the system minimum.
The interval between the LATEST-DELIV-DATE and EXPIRATION-DATE is the package expiration time window. You can specify the EXPIRATION-DATE as an absolute calendar date or a number of time units from LATEST-DELIV-DATE. An absolute or relative date is indicated by the REL-DATE-EXPIRATION field.

PRIORITY is the package priority. The field can have a value from 0 (lowest priority) to 199 (highest priority).

AGENT-SELECTOR is the agent selector criteria, as created by the ALTER-AGENT-SELECT UOW. This is a number that is stored with the package to determine which agents are invoked when the package is delivered. Refer to the ALTER-AGENT-SELECT UOW.

- RETN-CODE is the return code. TISERV returns a code in this field to indicate one of the following entries:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OK</td>
</tr>
<tr>
<td>4010</td>
<td>E-BAD-TRANSACTION</td>
</tr>
<tr>
<td>4035</td>
<td>E-ITEM-NOT-FOUND</td>
</tr>
<tr>
<td>4041</td>
<td>E-ITEM-UNALTERABLE</td>
</tr>
<tr>
<td>4042</td>
<td>E-ITEM-NOT-PKG-HDR</td>
</tr>
<tr>
<td>4050</td>
<td>E-UPDATE-MISMATCH</td>
</tr>
<tr>
<td>4051</td>
<td>E-MUST-BE-YN</td>
</tr>
<tr>
<td>4052</td>
<td>E-RESERVED-MUST-BE-N</td>
</tr>
<tr>
<td>4054</td>
<td>E-INVALID-AGENT-SEL</td>
</tr>
<tr>
<td>4056</td>
<td>E-INVALID-ITEM-TYPE</td>
</tr>
<tr>
<td>4072</td>
<td>E-UNITS-MUST-BE-DHM</td>
</tr>
<tr>
<td>4073</td>
<td>E-INVALID-DATE-TIME</td>
</tr>
<tr>
<td>4074</td>
<td>E-INVALID-REL-TIME-QTY</td>
</tr>
<tr>
<td>4078</td>
<td>E-INVALID-PRIORITY</td>
</tr>
<tr>
<td>4906</td>
<td>E-ERR-ITEMDESC-FILE</td>
</tr>
<tr>
<td>4912</td>
<td>E-ERR-FOLDER-FILE</td>
</tr>
<tr>
<td>4922</td>
<td>E-ERR-INV-FOLDER-FILE</td>
</tr>
</tbody>
</table>

- RETN-CODE-DETAIL is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

ALTER-ITEM-DESCR OPERATION. ALTER-ITEM-DESCR alters the fields in an item descriptor. This UOW selects for alteration those fields chosen in FIELDS-TO-ALTER, and modifies them in accordance with the information supplied in ALTERABLE-FIELDS.

If you do not issue this UOW for a package, the package will be delivered by TRANSFER in accordance with default values assumed at package creation.

If the item descriptor is not a package header item, ITEM-TYPE is the only field that can be altered.

If the UOW requests updates of multiple fields and one of the alterations is illegal and fails, none of the requested fields are updated.
ALTER-PROFILE-ELEM (UOW Code 204)

ALTER-PROFILE-ELEM alters data elements in a depot profile record. Typically, this UOW is issued after a GET-PROFILE-ELEM UOW.

Definition I (a DDL skeleton format to which definitions can be added; for general applications use):

```plaintext
DEF alter-profile-elem-usk.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 204.
  02 filler TYPE CHARACTER 4.
  02 corr-name PIC X(80) VALUE SPACES.
  02 num-returned TYPE BINARY 16 UNSIGNED.
END.

DEF alter-profile-elem-rsk.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 204.
    02 retn-code TYPE BINARY 16.
    02 retn-code-detail TYPE BINARY 16.
    02 corr-name PIC X(80).
    02 num-returned TYPE BINARY 16.
END.
```

Definition II (for TAL programs):

```plaintext
DEF alter-profile-elem-uow.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 204.
  02 filler TYPE CHARACTER 4.
  02 corr-name PIC X(80) VALUE SPACES.
  02 num-returned TYPE BINARY 16 UNSIGNED.
  02 elem-data-block TYPE PROFILE-ELEM-SHORT OCCURS 0 TO 10 TIMES DEPENDING ON num-returned.
END.
```

See GET-PROFILE-ELEM for Operational Details
UOW Descriptions
ALTER-PROFILE-ELEM

DEF alter-profile-elem-rsp.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 204.
  02 retn-code TYPE BINARY 16.
  02 retn-code-detail TYPE BINARY 16.
  02 corr-name PIC X(80).
  02 num-returned TYPE BINARY 16 UNSIGNED.
  02 elem-retn-code TYPE BINARY 16 OCCURS 0 TO 10 TIMES DEPENDING ON num-returned.
END.

ALTER-PROFILE-ELEM FIELDS. The fields defined in this UOW are:

- **HDR** is the UOW header. The UOW-CODE value is 204.

- **CORR-NAME** is the name of the correspondent whose depot profile will be altered. This is also the correspondent represented by the calling process. Only system administrators can enter correspondent names other than their own in this field; correspondents without system administrator privileges must enter their own names or leave the field blank. On input, you can use a partially qualified name, including wildcard characters. In the response, the fully qualified name is returned.

- **NUM-RETURNED** is the number of data elements that you will alter. You can specify any number of elements, restricted only by the length of the entire IPC that contains this UOW. In the response, this field contains:
  - the number actually altered if no serious errors are detected
  - zero if serious errors are detected

- **ELEM-DATA-BLOCK** (used in Definition II only) represents specific fields to be altered, and is either 36 or 84 bytes long. The format of the ELEM-DATA-BLOCK field is the same as the format in which this field is returned by the GET-PROFILE-ELEM UOW. Refer to the discussion of the GET-PROFILE-ELEM UOW for details.

See GET-PROFILE-ELEM for Operational Details
You can specify several element blocks in one ALTER-PROFILE-ELEM UOW. Long and short element blocks can, in fact, be mixed; TISERV accepts a mixture of block lengths in one occurrence of this UOW.

For general applications (Definition I), you typically construct an ALTER-PROFILE-ELEM UOW by specifying all fields up to and including NUM-RETURNED, and then specifying separate definitions for each particular element. Alternatively, you can define the element block as a DEPENDING ON construction, using either PROFILE-ELEMENT-SHORT or PROFILE-ELEMENT-LONG; notice, however, that this type of construction does not permit mixing both long and short blocks. An example of an element block construction is:

```
DEF 02 my-special-request TYPE ALTER-PROFILE-ELEMENT-UOW.
02 data-depot-priorities TYPE *.
02 data-depot-mail-flags TYPE *.
02 data-depot-mail-filename TYPE *.
END.
```

For this example, NUM-RETURNED would be set to 3.

- RETN-CODE is the return code. TISERV returns a code in this field to indicate one of the following entries.

  To indicate successful record alteration:

  0 OK

  To indicate problems with the correspondent name:

  5600 E-CORR-NSRV-ERR
  5601 E-CORR-NOT-FOUND
  5602 E-CORR-BAD-NAME
  5604 E-CORR-NO-SUCH-NODE
  5606 E-CORR-NSRV-NOT-FOUND

  To indicate other problems:

  4010 E-BAD-TRANSACTION
  4093 E-SECURITY-VIOLATION
  4201 E-CONTEXT-ERR
  4210 W-IDENTIFIER-ERRS
  4902 E-ERR-PROFILE-FILE

- RETN-CODE-DETAIL is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

- ELEM-RETN-CODE (used in Definition II only) supplies return codes for specific data elements. The response contains the same number of return codes as the number of elements for which alteration was requested.

See GET-PROFILE-ELEM for Operational Details
UOW Descriptions
ALTER-PROFILE-ELEM

To indicate successful alteration of the element:

0 OK

To indicate errors encountered with the element, see the discussion of the GET-PROFILE-ELEM UOW.

ALTER-PROFILE-ELEM OPERATION. The ALTER-PROFILE-ELEM UOW and the GET-PROFILE-ELEM UOW complement each other. Because these two UOWs are closely involved with one another and because they share common field definitions, the details of their operation are discussed in a common area of this manual, under the description of the GET-PROFILE-ELEM UOW. Refer to that description for further information about the ALTER-PROFILE-ELEM UOW.

The GET-PROFILE-ELEM response has exactly the same format as the ALTER-PROFILE-ELEM request. To use the same definition for both UOWs, you only need to change the UOW code value.
ATTACH-COMPNT-A01 (UOW Code 136)

ATTACH-COMPNT-A01 attaches a component item to a parent.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hdr</td>
<td>UOW header</td>
</tr>
<tr>
<td>self-ident</td>
<td>UOW CODE value is &quot;UW&quot;</td>
</tr>
<tr>
<td>uow-code</td>
<td>TYPE BINARY 16 UNSIGNED VALUE 136</td>
</tr>
<tr>
<td>parent-item-id</td>
<td>PARENT-ITEM-ID is the parent item</td>
</tr>
<tr>
<td>compnt-id</td>
<td>COMPNT-ID is the component item</td>
</tr>
<tr>
<td>rel-position</td>
<td>REL-POSITION is the position that the component item will occupy in the</td>
</tr>
<tr>
<td></td>
<td>parent item component list. Unless this is the last position, the entry</td>
</tr>
<tr>
<td></td>
<td>previously occupying this position and all entries following it are shifted</td>
</tr>
<tr>
<td></td>
<td>to the next higher position. The first component item occupies relative</td>
</tr>
<tr>
<td></td>
<td>position 1. The value 0 is illegal in this field.</td>
</tr>
</tbody>
</table>

ATTACH-COMPNT-A01 FIELDS. The fields defined in this UOW are:

- HDR is the UOW header. The UOW-CODE value is 136.
- PARENT-ITEM-ID is the parent item.
- COMPNT-ID is the component item.
- REL-POSITION is the position that the component item will occupy in the parent item component list. Unless this is the last position, the entry previously occupying this position and all entries following it are shifted to the next higher position. The first component item occupies relative position 1. The value 0 is illegal in this field.
If the relative position of the last component is unknown, you can reference this position by setting the REL-POSITION field to -1. The value -1 means the last position of the list. Suppose, for example, that the list contained 30 components. If you issued the ATTACH-COMPNT-A01 UOW to attach a new component to the item and used the value -1 in REL-POSITION, that new component would occupy the 31st position in the list, as illustrated in Figure 5-3.

Figure 5-3. Attaching a New Component to a Parent Item

- COMPNT-TYPE is a numeric value that TRANSFER will save with the component. The value can range from 0 through 9999; values 100 through 999 are reserved for Tandem. You can use this field to indicate why the component was attached. This field is returned with the component by the GET-ITEM-COMPNT-A01 UOW. The field is not interpreted by TRANSFER.
• **RETN-CODE** is the return code. TISERV returns a code in this field to indicate one of the following entries:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OK</td>
<td>4039</td>
<td>E-COMPNT-NOT-FOUND</td>
</tr>
<tr>
<td>4010</td>
<td>E-BAD-TRANSACTION</td>
<td>4040</td>
<td>E-BAD-ITEM-DESCR</td>
</tr>
<tr>
<td>4036</td>
<td>E-ITEM-TOO-COMPLEX</td>
<td>4041</td>
<td>E-ITEM-UNALTERABLE</td>
</tr>
<tr>
<td>4037</td>
<td>E-PARENT-NOT-FOUND</td>
<td>4057</td>
<td>E-INVALID-REL-POSITION</td>
</tr>
<tr>
<td>4038</td>
<td>E-COMPNT-CYCLE</td>
<td>4096</td>
<td>E-INVALID-COMPNT-TYPE</td>
</tr>
</tbody>
</table>

• **RETN-CODE-DETAIL** is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

• **INSERTED-POSITION** is the position actually occupied by the new item after insertion into the components list.

ATTACH-COMPNT-AOl OPERATION. ATTACH-COMPNT-AOl attaches one item as a component of another. TISERV records the item IDs of components in components list records that are treated as part of the parent item. If this operation is successful, the component count in the parent item descriptor, and the parent count in the component item descriptor are both incremented by 1.

A component item can contain its own components, each of which can also have components; thus, you can create an entire tree of items. The parent item cannot itself be in its own tree of components; this is known as a component cycle as described in Section 6.
CANCEL-PKG (UOW Code 118)

CANCEL-PKG cancels a package. A package can only be canceled by its sender.

**CANCEL-PKG FIELDS.** The fields defined in this UOW are:

- **HDR** is the UOW header. The UOW-CODE value is 118.

- **ITEM-ID** is the item ID of the package header for the package to be canceled.

- **RETN-CODE** is the return code. TISERV returns a code in this field to indicate one of the following entries:

  - 0 OK
  - 4010 E-BAD- TRANSACTION
  - 4035 E-ITEM-NOT- FOUND
  - 4042 E-ITEM-NOT-PKG-HDR
  - 4044 E-PREVIOUSLY- CANCELED
  - 4045 E-TSCHED- UNAVAIL
  - 4081 W-DELIV-IN-PROGRESS
  - 4083 E-NOT-CREATED-BY-YOU
  - 4084 E-PKG-NOT-SUBMITTED

- **RETN-CODE-DETAIL** is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.
CANCEL-PKG OPERATION. CANCEL-PKG cancels delivery of the package indicated by ITEM-ID. TRANSFER removes this package from the INBOX folder of those recipients that have not acknowledged receipt. The CANCEL-PKG UOW also prevents any future deliveries of the package.

TRANSFER sends a cancellation notification package to any recipients who have already examined the package with the ACK-RECEIPT UOW. In addition, TRANSFER notifies the sender of any recipient who has examined the canceled package.

NOTE

Cancellation of packages after their delivery has been initiated might create inconsistencies at the application level, and might not fully eliminate all traces of the package in the system or network.
COPY-ITEM (UOW Code 107)

COPY-ITEM makes a copy of an existing item.

COPY-ITEM FIELDS. The fields defined in this UOW are:

- HDR is the UOW header. The UOW-CODE value is 107.
- ITEM-ID is the item to be copied (the source item).
- OPTIONS allow you to specify what is included in the copying operation by entering Y (for yes) or N (for no):
  - COPY-DATA - copy all data records associated with the item.
  - COPY-RECIPS - copy all recipients associated with the item if the item is a package header item.
COPY-COMPNTS - attach all components of the original item to the new copy of the item.

RESERVED-3 - reserved for use by Tandem; this field must be set to N.

- RETN-CODE is the return code. TISERV returns a code in this field to indicate one of the following entries:

  0  OK
  4010 E-BAD-TRANSACTION  4051 E-MUST-BE-YN
  4035 E-ITEM-NOT-FOUND  4052 E-RESERVED-MUST-BE-N

- RETN-CODE-DETAIL is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

- NEW-ITEM-ID is the ID of the new item created by the copy operation.

COPY-ITEM OPERATION. COPY-ITEM makes a copy of an existing item. This creates a new item. If the COPY-DATA field is set to Y, the new item contains the same client data as the item specified by ITEM-ID. This operation is equivalent to issuing a CREATE-ITEM request, followed by one ADD-ITEM-REC request for each client record in the original item. The new item is identified by the ID returned in NEW-ITEM-ID.

If the item is a package header, TRANSFER copies into the new item descriptor those fields that can be modified by your client; these are the fields that can be specified in the ALTER-ITEM-DESCR UOW. All other descriptor fields in the copy operation contain default values assigned by TRANSFER.

If COPY-RECIPS = Y TRANSFER copies all recipients associated with the item if the item is a package header item.

If COPY-COMPNTS = Y TRANSFER attaches all components of the original item to the copied item, and increments the parent count of each component by 1.

If you must retain access to the new item beyond the end of the current session, you must either save that item in a folder or attach the item to another item. The new item is saved in an internal temporary folder upon creation and is removed from that folder when the session is terminated.
UOW Descriptions
CREATE-DEPOT

CREATE-DEPOT (UOW Code 201)

CREATE-DEPOT creates a new correspondent and depot. This UOW can be issued only by correspondents with system administrator capability.

```
DEF create-depot-uow.
  02 hdr.
      03 self-ident PIC AA VALUE "UW".
      03 uow-code TYPE BINARY 16 UNSIGNED VALUE 201.
  02 new-corr PIC X(80).
  02 model-name PIC X(80) VALUE SPACES.
END.
```

```
DEF create-depot-rsp.
  02 hdr.
      03 self-ident PIC AA VALUE "UW".
      03 uow-code TYPE BINARY 16 UNSIGNED VALUE 201.
  02 retn-code TYPE BINARY 16.
  02 retn-code-detail TYPE BINARY 16.
END.
```

CREATE-DEPOT FIELDS. The fields defined in this UOW are:

- **HDR** is the UOW header. The UOW-CODE value is 201.

- **NEW-CORR** is the name of the new correspondent.

- **MODEL-NAME** is the name of an existing correspondent whose depot will be used as a model depot. The profile and the special folders for the model depot are used as a model to construct the profile and the special folders for the new depot. TRANSFER copies the attributes (all profile records and special folder ordering criteria) of the model depot into the depot for the new correspondent. If you do not specify a model name in this field, TRANSFER uses the default model depot.

- **RETN-CODE** is the return code. TISERV returns a code in this field to indicate one of the following entries.
To indicate successful creation of the new correspondent and depot:

0 OK

To indicate problems with the new correspondent name:

- 4208 E-CORR-BLANK
- 5600 E-CORR-NSRV-ERR
- 5602 E-CORR-BAD-NAME
- 5606 E-CORR-NSRV-NOT-FOUND

To indicate problems with the model name:

- 4205 E-MODEL-DEPOT-ABSENT
- 5725 E-MODEL-NSRV-ERR
- 5726 E-MODEL-NOT-FOUND
- 5727 E-MODEL-BAD-NAME
- 5729 E-MODEL-NO-SUCH-NODE

To indicate other problems:

- 4010 E-BAD-TRANSACTION
- 4093 E-SECURITY-VIOLATION

- RETN-CODE-DETAIL is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

CREATE-DEPOT OPERATION. The CREATE-DEPOT UOW adds a new correspondent to the system by entering the correspondent name in the TRANSFER name directory, and creates a depot for that correspondent. The depot is created on the same node as the copy of TISERV accessed by this UOW.

If a depot already exists for the correspondent identified by NEW-CORR or if certain other errors occur, TRANSFER does not create the depot.

If you specify a model depot in the MODEL-NAME field, TRANSFER copies all profiles, including any agents, from the model depot to the new depot profile. TRANSFER uses the ordering criteria that was specified for the special folders in the model depot for the special folders in the new depot. If you do not specify a model depot, TRANSFER uses the default model depot.
UOW Descriptions
CREATE-DLIST

CREATE-DLIST (UOW Code 217)

CREATE-DLIST creates a distribution list.

```
DEF create-dlist-uow.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 217.
    02 corr-name PIC X(80) VALUE SPACES.
    02 dlist-name PIC X(80).
END.
```

```
DEF create-dlist-rsp.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 217.
    02 retn-code TYPE BINARY 16.
    02 retn-code-detail TYPE BINARY 16.
    02 corr-name PIC X(80).
END.
```

CREATE-DLIST FIELDS. The fields defined in this UOW are:

- HDR is the UOW header. The UOW-CODE value is 217.
- CORR-NAME is the name of the correspondent represented by the calling process. Only system administrators can enter correspondent names other than their own in this field; correspondents without system administrator privileges must enter their own names or leave the field blank. On input, you can use a partially qualified name, including wildcard characters. In the response, the fully qualified name is returned.
- DLIST-NAME is the name to be assigned to the new distribution list. The name must be either a TRANSFER simple name or a fully qualified name.
- RETN-CODE is the return code. TISERV returns a code in this field to indicate one of the following entries.

  To indicate successful creation of the distribution list:

  0  OK
To indicate problems with the correspondent name:

5600  E-CORR-NSRV-ERR
5601  E-CORR-NOT-FOUND
5602  E-CORR-BAD-NAME
5604  E-CORR-NO-SUCH-NODE
5606  E-CORR-NSRV-NOT-FOUND
5607  E-CORR-NSRV-DOWN
5611  E-CORR-NET-DOWN
5622  E-CORR-NOT-SAME-NODE
5623  E-CORR-AMBIGUOUS-NAME

To indicate problems with the distribution list name:

5625  E-DLIST-NSRV-ERR
5627  E-DLIST-BAD-NAME
5629  E-DLIST-NO-SUCH-NODE
5631  E-DLIST-NSRV-NOT-FOUND
5632  E-DLIST-NSRV-DOWN
5633  E-DLIST-NO-PARENT
5636  E-DLIST-NET-DOWN
5637  E-DLIST-ALREADY EXISTS
5647  E-DLIST-NOT-SAME-NODE
5648  E-DLIST-AMBIGUOUS-NAME

To indicate other problems:

4010  E-BAD-TRANSACTION
4093  E-SECURITY-VIOLATION
4201  E-CONTEXT-ERR
4902  E-ERR-PROFILE-FILE

• RETN-CODE-DETAIL is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

CREATE-DLIST OPERATION. CREATE-DLIST creates a new distribution list with the name assigned in DLIST-NAME.
CREATE-FOLDER (UOW Code 227)

CREATE-FOLDER creates a new folder. This UOW is maintained for version compatibility only. CREATE-FOLDER-B00 is the recommended UOW.

```
DEF create-folder-uow.
  02 hdr.
    03 self-ident          PIC AA VALUE "UW".
    03 uow-code            TYPE BINARY 16 UNSIGNED VALUE 227.
    02 corr-name           PIC X(80) VALUE SPACES.
    02 folder-name         PIC X(80).
END.

DEF create-folder-rsp.
  02 hdr.
    03 self-ident          PIC AA VALUE "UW".
    03 uow-code            TYPE BINARY 16 UNSIGNED VALUE 227.
    02 retn-code           TYPE BINARY 16.
    02 retn-code-detail     TYPE BINARY 16.
    02 corr-name           PIC X(80).
END.
```

CREATE-FOLDER FIELDS. The fields defined in this UOW are:

- **HDR** is the UOW header. The UOW-CODE value is 227.
- **CORR-NAME** is the name of the correspondent represented by the calling process. Only system administrators can enter correspondent names other than their own in this field; correspondents without system administrator privileges must enter their own names or leave the field blank. On input, you can use a partially qualified name, including wildcard characters. In the response, the fully qualified correspondent name is returned.
- **FOLDER-NAME** is the name to be assigned to the new folder. The name must be either a TRANSFER simple name or a fully qualified name.
- **RETN-CODE** is the return code. TISERV returns a code in this field to indicate one of the following entries.
To indicate successful creation of the folder:

0 OK

To indicate problems with the correspondent name:

5600 E-CORR-NSRV-ERR
5601 E-CORR-NOT-FOUND
5602 E-CORR-BAD-NAME
5604 E-CORR-NO-SUCH-NODE
5606 E-CORR-NSRV-NOT-FOUND

To indicate problems with the folder name:

5675 E-FLD-NSRV-ERR
5677 E-FLD-BAD-NAME
5679 E-FLD-NO-SUCH-NODE
5681 E-FLD-NSRV-NOT-FOUND
5682 E-FLD-NSRV-DOWN

To indicate other problems:

4010 E-BAD-TRANSACTION
4093 E-SECURITY-VIOLATION
4201 E-CONTEXT-ERR
4902 E-ERR-PROFILE-FILE

• RETN-CODE-DETAIL is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

CREATE-FOLDER OPERATION. CREATE-FOLDER creates a new folder with the name assigned in FOLDER-NAME. The folder is created in ascending sequence by time saved, which is the ordering criteria used by TRANSFER A01/A02.
CREATE-FOLDER-B00 (UOW Code 232)

CREATE-FOLDER-B00 creates a new folder and establishes the ordering criteria by which items will be stored in the folder.

**UOW Descriptions**

**CREATE-FOLDER-B00**

**CREATE-FOLDER-B00 FIELDS.** The fields defined in this UOW are:

- HDR is the UOW header. The UOW-CODE value is 232.

```plaintext
DEF create-folder-b00-uow.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 232.
    02 corr-name PIC X(80) VALUE SPACES.
    02 folder-name PIC X(80).
    02 ordering-discipline TYPE CHARACTER 1.
      88 time-saved VALUE "T".
      88 creator-name VALUE "C".
      88 earliest-deliv-date VALUE "E".
      88 applic-defined VALUE "A".
    02 filler PIC X(1) VALUE SPACES.
    02 applic-order-type PIC 9(4) COMP.
    02 options.
      03 ascending-sequence TYPE BOOLEAN.
      03 allow-duplicates TYPE BOOLEAN.
      03 reserved-2 TYPE BOOLEAN VALUE "N".
      03 reserved-3 TYPE BOOLEAN VALUE "N".
      03 reserved-4 TYPE BOOLEAN VALUE "N".
      03 reserved-5 TYPE BOOLEAN VALUE "N".
      03 reserved-6 TYPE BOOLEAN VALUE "N".
      03 reserved-7 TYPE BOOLEAN VALUE "N".
END.

DEF create-folder-b00-rsp.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 232.
    02 retn-code TYPE BINARY 16.
    02 retn-code-detail TYPE BINARY 16.
    02 corr-name PIC X(80).
END.
```
• **CORR-NAME** is the name of the correspondent represented by the calling process. Only system administrators can enter correspondent names other than their own in this field; correspondents without system administrator privileges must enter their own names or leave the field blank. On input, you can use a partially qualified name, including wildcard characters. In the response, the fully qualified correspondent name is returned.

• **FOLDER-NAME** is the name to be assigned to the new folder. The name must be either a TRANSFER simple name or a fully qualified name.

• **ORDERING-DISCIPLINE** specifies what information TISERV should use as the ordering key when saving items in the folder as follows:

<table>
<thead>
<tr>
<th>Entry</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIME-SAVED (T)</td>
<td>Save items in chronological order of the time a SAVE-ITEM UOW is executed for the items.</td>
</tr>
<tr>
<td>CREATOR-NAME (C)</td>
<td>Save items in alphabetic order by creator name.</td>
</tr>
<tr>
<td>EARLIEST-DELIVERY DATE (E)</td>
<td>Save items that are package headers and are unalterable by earliest delivery date; save items that are not package headers or are package headers and are alterable by creation date.</td>
</tr>
<tr>
<td>APPLIC-DEFINED (A)</td>
<td>Save items in order of the ordering key specified by the application in the SAVE-ITEM-BY-KEY UOW. An application can specify the same ordering key for multiple items if duplicates are allowed; if duplicates are not allowed, the application is responsible for ensuring the uniqueness of the key.</td>
</tr>
</tbody>
</table>

If this field contains a character other than T, C, E, or A, TISERV returns the error E-INVALID-ORD-DISCIPLN.
UOW Descriptions
CREATE-FOLDER-B00

- APPLIC-ORDER-TYPE is meaningful only if ORDERING-DISCIPLINE is APPLIC-DEFINED (A). If ORDERING-DISCIPLINE is not APPLIC-DEFINED, the field is ignored.

This field is not interpreted by TRANSFER, but is stored as part of the folder's ordering criteria and is returned by the GET-FOLDER-ORDER UOW. This enables a TRANSFER application to have several types of APPLIC-DEFINED ordering; the APPLIC-ORDER-TYPE can be used to distinguish between types.

The field can have a value from 0 through 9999; values 100 through 999 are reserved for Tandem. If a number outside this range is specified, TISERV returns the error E-INTEGRATE-APPLIC-ORDER-TYPE.

- OPTIONS provides additional criteria for saving items. Two options are provided.

  ASCENDING-SEQUENCE determines whether items are saved within the folder in ascending or descending key order.

  Y = Save items in ascending key order.

  N = Save items in descending key order.

Variable length APPLIC-DEFINED keys that are used with descending key order will not collate in the correct order unless the application pads the key with enough bytes to make it a constant length for all entries in the folder.

A folder ordered by CREATOR-NAME in descending sequence is an example of a field that is variable length and is automatically padded to be 120 bytes. For a folder ordered by CREATOR-NAME in ascending sequence, the amount of disc space used corresponds directly to the length of creator names and the number of items saved in a folder. Considering creator names of a maximum of 20 characters, the cost of descending sequence over ascending sequence in terms of disc space is approximately 100 bytes per item saved in a folder ordered by CREATOR-NAME.

ALLOW-DUPLICATES determines whether or not multiple items with the same ordering key can be saved in a folder.

Y = Duplicate ordering keys are allowed.

N = Duplicate ordering keys are not allowed. This field has no effect if the ORDERING-DISCIPLINE is TIME-SAVED (T); TIME-SAVED ordering already ensures a unique key.
RESERVED-2 through RESERVED-7 are reserved for use by Tandem; these fields must be set to N.

- RETN-CODE is the return code. TISERV returns a code in this field to indicate one of the following entries.

  To indicate successful creation of the folder:

  0 OK

  To indicate problems with the correspondent name:

  5600 E-CORR-NSRV-ERR 5607 E-CORR-NSRV-DOWN
  5601 E-CORR-NOT-FOUND 5611 E-CORR-NET-DOWN
  5602 E-CORR-BAD-NAME 5622 E-CORR-NOT-SAME-NODE
  5604 E-CORR-NO-SUCH-NODE 5623 E-CORR-AMBIGUOUS-NAME
  5606 E-CORR-NSRV-NOT-FOUND

  To indicate problems with the folder name:

  5675 E-FLD-NSRV-ERR 5683 E-FLD-NO-PARENT
  5677 E-FLD-BAD-NAME 5686 E-FLD-NET-DOWN
  5679 E-FLD-NO-SUCH-NODE 5687 E-FLD-ALREADY- EXISTS
  5681 E-FLD-NSRV-NOT-FOUND 5697 E-FLD-NOT-SAME-NODE
  5682 E-FLD-NSRV-DOWN 5698 E-FLD-AMBIGUOUS-NAME

  To indicate other problems:

  4010 E-BAD TRANSACTION 4265 E-INVALID-ORD-DISCIPLN
  4051 E-MUST-BE-YN 4266 E-INVALID-APP-ORD-TYPE
  4052 E-RESERVED-MUST-BE-N 4902 E-ERR-PROFILE-FILE
  4093 E-SECURITY-VIOLATION 4904 E-ERR-SESSION-FILE
  4201 E-CONTEXT-ERR

- RETN-CODE-DETAIL is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

CREATE-FOLDER-B00 OPERATION. CREATE-FOLDER-B00 creates a new folder with the specified ordering criteria.

Folder order affects the order in which items are saved and subsequently referenced. SCAN-FOLDER, SCAN-FOLDER-B00, and SCAN-FOLDER-BY-KEY UOWs return items according to the ordering key by which items are saved in the folder.
CREATE-ITEM (UOW Code 103)

CREATE-ITEM creates an item.

```
DEF create-item-uow.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 103.
  02 item-type PIC 9(4) COMP.
  02 is-pkg-hdr TYPE BOOLEAN.
  02 reserved-1 TYPE BOOLEAN VALUE "N".
END.
```

```
DEF create-item-rsp.
  02 hdr TYPE UOW-HDR.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 103.
  02 retn-code TYPE BINARY 16.
  02 retn-code-detail TYPE BINARY 16.
  02 item-id.
    03 dummy PIC X(12).
END.
```

CREATE-ITEM FIELDS. The fields defined in this UOW are:

- HDR is the UOW header. The UOW-CODE value is 103.
- ITEM-TYPE is a numeric value that lets you categorize items by type. Its meaning is defined by your application. The value must be in the range of 0-9999; values 100 through 999 are reserved for use by Tandem.

**NOTE**

ITEM-TYPE should not be confused with RECORD-TYPE, which has a totally different meaning.

- IS-PKG-HDR determines whether the new item is a package header.

  Y = The item is a package header.

  N = The item is not a package header.
- RESERVED-1 is reserved for use by Tandem; this field must be set to N.

- RETN-CODE is the return code. TISERV returns a code in this field to indicate one of the following entries:

```
0  OK
4010  E-BAD-TRANSACTION
4051  E-MUST-BE-YN
```

- RETN-CODE-DETAIL is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

- ITEM-ID is the ID assigned to the new item and returned by TRANSFER.

CREATE-ITEM OPERATION. CREATE-ITEM creates a new item and returns the ID in the ITEM-ID field. If the item is to be a package header, as specified in the IS-PKG-HDR field, TRANSFER appends default package header information to the item descriptor.

When the item is created, it is saved in an internal temporary folder and is removed from that folder when the session is terminated. If you must retain access to the new item beyond the end of the current session, you must either save the item in a folder or attach the item to another item.
DELETE-DEPOT (UOW Code 202)

DELETE-DEPOT deletes a correspondent and depot. This UOW can be issued only by correspondents with system administrator write privileges.

DEF delete-depot-uow.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 202.
  02 corr-name PIC X(80).
  02 force-flag TYPE CHARACTER 1.
    88 forced-deletion VALUE "Y".
    88 do-not-force VALUE "N".
  02 filler TYPE CHARACTER 1 VALUE SPACES.
END.

DEF delete-depot-rsp.
  02 hdr TYPE UOW-HDR.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 202.
  02 retn-code TYPE BINARY 16.
  02 retn-code-detail TYPE BINARY 16.
  02 corr-name PIC X(80).
END.

DELETE-DEPOT FIELDS. The fields defined in this UOW are:

- HDR is the UOW header. The UOW-CODE value is 202.
- CORR-NAME is the name of the correspondent whose depot is to be deleted. You can use a partially qualified name, including wildcard characters. In the response, the fully qualified name is returned.
- FORCE-FLAG determines whether TISERV forces deletion even if the depot folders contain items or packages or if any distribution lists defined for the depot contain members.
  
  Y = Force deletion.

  N = Do not force deletion. If the depot folders contain items or packages or if any depot distribution lists contain members, the depot is not deleted, an error indication is returned, and you should abort the
transaction. If distribution lists and folders are all empty, TISERV deletes the depot.

- RETN-CODE is the return code. TISERV returns a code in this field to indicate one of the following entries.

  To indicate successful record alteration:
  
  0 OK

  To indicate problems with the correspondent name:

  5600 E-CORR-NSRV-ERR
  5601 E-CORR-NOT-FOUND
  5602 E-CORR-BAD-NAME
  5604 E-CORR-NO-SUCH-NODE
  5606 E-CORR-NSRV-NOT-FOUND

  5607 E-CORR-NSRV-DOWN
  5611 E-CORR-AMBIGUOUS-NAME
  5622 E-CORR-NOT-SAME-NODE
  5623 E-CORR-AMBIGUOUS-NAME

  Objects belonging to correspondents are also deleted; therefore, a similar set of error messages for folders and distribution lists can be received. Messages for folders begin with E-FLD- and messages for distribution lists begin with E-DLIST-.

  To indicate other problems:

  4010 E-BAD-TRANSACTION
  4051 E-MUST-BE-YN
  4093 E-SECURITY-VIOLATION
  4201 E-CONTEXT-ERR
  4213 E-ITEMS-EXIST

  4219 E-SESSION-ACTIVE
  4225 E-MEMBERS-EXIST
  4902 E-ERR-PROFILE-FILE
  4912 E-ERR-FOLDER-FILE
  4914 E-ERR-DLIST-FILE

  RETN-CODE-DETAIL is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

DELETE-DEPOT OPERATION. DELETE-DEPOT removes all profile records relating to the correspondent identified by CORR-NAME, and also removes the correspondent name from the TRANSFER name directory; this effectively deletes the depot of the correspondent. In addition, TRANSFER removes any distribution lists and folders relating to the depot.

NOTE

Deletion does not occur if a session is active at the depot when this UOW is issued.
DELETE-DLIST

DELETE-DLIST (UOW Code 218)

DELETE-DLIST deletes a distribution list.

```
DEF delete-dlist-uow.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 218.
  02 corr-name PIC X(80) VALUE SPACES.
  02 dlist-name PIC X(80).
  02 force-flag TYPE CHARACTER 1.
    88 forced-deletion VALUE "Y".
    88 do-not-force VALUE "N".
  02 filler TYPE CHARACTER 1.
END.
```

```
DEF delete-dlist-rsp.
  02 hdr TYPE UOW-HDR.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 218.
    02 retn-code TYPE BINARY 16.
    02 retn-code-detail TYPE BINARY 16.
    02 corr-name PIC X(80).
    02 dlist-name PIC X(80).
END.
```

DELETE-DLIST FIELDS. The fields defined in this UOW are:

- HDR is the UOW header. The UOW-CODE value is 218.

- CORR-NAME is the name of the correspondent represented by the calling process. Only system administrators can enter correspondent names other than their own in this field; correspondents without system administrator privileges must enter their own names or leave the field blank. You can use a partially qualified name, including wildcard characters. In the response, the fully qualified name is returned.

- DLIST-NAME is the name of the distribution list to be deleted. This can be a partially qualified name, including wildcard characters. In the response, the fully qualified name is returned.
• FORCE-FLAG determines whether TISERV forces deletion even if the distribution list contains members.

Y = Force deletion.

N = Do not force deletion. If the distribution list contains members, the distribution list is not deleted; an error indication is returned, and you should abort the transaction.

• RETN-CODE is the return code. TISERV returns a code in this field to indicate one of the following entries.

To indicate successful deletion:

0  OK

To indicate problems with the correspondent name:

5600  E-CORR-NSRV-ERR  5607  E-CORR-NSRV-DOWN
5601  E-CORR-NOT-FOUND  5611  E-CORR-NET-DOWN
5602  E-CORR-BAD-NAME   5622  E-CORR-NOT-SAME-NODE
5604  E-CORR-NO-SUCH-NAME  5623  E-CORR-AMBIGUOUS-NAME
5606  E-CORR-NSRV-NOT-FOUND

To indicate problems with the distribution list name:

5625  E-DLIST-NSRV-ERR  5632  E-DLIST-NSRV-DOWN
5626  E-DLIST-NOT-FOUND  5636  E-DLIST-NET-DOWN
5627  E-DLIST-BAD-NAME   5647  E-DLIST-NOT-SAME-NODE
5629  E-DLIST-NO-SUCH-NAME  5648  E-DLIST-AMBIGUOUS-NAME
5631  E-DLIST-NSRV-NOT-FOUND

To indicate other problems:

4010  E-BAD-TRANSACTION  4225  E-MEMBERS-EXIST
4051  E-MUST-BE-YN  4902  E-ERR-PROFILE-FILE
4093  E-SECURITY-VIOLATION  4914  E-ERR-DLIST-FILE
4201  E-CONTEXT-ERR

• RETN-CODE-DETAIL is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

DELETE-DLIST OPERATION. DELETE-DLIST deletes a distribution list by first deleting all members from the list (if FORCE-FLAG is set to Y), and then removing the list name from the TRANSFER name directory.

Any pointers from existing lists to a deleted distribution list remain in the system, but have no meaning. To provide good housekeeping, you should purge these pointers.
DELETE-FOLDER (UOW Code 230)

DELETE-FOLDER deletes a folder.

**DEF delete-folder-uow.**

```plaintext
02 hdr.
  03 self-ident PIC AA VALUE "UW".
  03 uow-code TYPE BINARY 16 UNSIGNED VALUE 230.
  02 corr-name PIC X(80) VALUE SPACES.
  02 folder-name PIC X(80).
  02 force-flag TYPE CHARACTER 1.
    88 forced-deletion VALUE "Y".
    88 do-not-force VALUE "N".
  02 filler TYPE CHARACTER 1 VALUE SPACES.
END.
```

```plaintext
DEF delete-folder-rsp.

02 hdr.
  03 self-ident PIC AA VALUE "UW".
  03 uow-code TYPE BINARY 16 UNSIGNED VALUE 230.
  02 retn-code TYPE BINARY 16.
  02 retn-code-detail TYPE BINARY 16.
  02 corr-name PIC X(80).
  02 folder-name PIC X(80).
END.
```

**DELETE-FOLDER FIELDS.** The fields defined in this UOW are:

- **HDR** is the UOW header. The UOW-CODE value is 230.

- **CORR-NAME** is the name of the correspondent represented by the calling process. Only system administrators can enter correspondent names other than their own in this field; correspondents without system administrator privileges must enter their own names or leave the field blank. On input, you can use a partially qualified name, including wildcard characters. In the response, the fully qualified name is returned.

- **FOLDER-NAME** is the name of the folder to be deleted. This can be a partially qualified name, including wildcard characters. In the response, the fully qualified name is returned.
• FORCE-FLAG determines whether TISERV forces the deletion even if the folder contains packages or items.

  Y = Force deletion.

  N = Do not force deletion. If the folder contains packages or items, the folder is not deleted; an error indication is returned, and you should abort the transaction.

• RETN-CODE is the return code. TISERV returns a code in this field to indicate one of the following entries.

  To indicate successful deletion:

  0 OK

  To indicate problems with the correspondent name:

  5600 E-CORR-NSRV-ERR  5607 E-CORR-NSRV-DOWN
  5601 E-CORR-NOT-FOUND  5611 E-CORR-NET-DOWN
  5602 E-CORR-BAD-NAME    5622 E-CORR-NOT-SAME-NODE
  5604 E-CORR-NO-SUCH-NODE 5623 E-CORR-AMBIGUOUS-NAME
  5606 E-CORR-NSRV-NOT-FOUND

  To indicate problems with the folder name:

  5675 E-FLD-NSRV-ERR    5682 E-FLD-NSRV-DOWN
  5676 E-FLD-NOT-FOUND   5686 E-FLD-NET-DOWN
  5677 E-FLD-BAD-NAME    5697 E-FLD-NOT-SAME-NODE
  5679 E-FLD-NO-SUCH-NODE 5698 E-FLD-AMBIGUOUS-NAME
  5681 E-FLD-NSRV-NOT-FOUND

  To indicate other problems:

  4010 E-BAD-TRANSACTION  4213 E-ITEMS-EXIST
  4051 E-MUST-BE-YN       4218 W-CONTENTS-PURGED
  4093 E-SECURITY-VIOLATION 4902 E-ERR-PROFILE-PURGED
  4105 E-CONCURRNT-FLD-UPDATE 4912 E-ERR-FOLDER-FILE
  4201 E-CONTEXT-ERR     4922 E-ERR-INV-FOLDER-FILE

• RETN-CODE-DETAIL is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

DELETE-FOLDER OPERATION. DELETE-FOLDER deletes the folder identified by FOLDER-NAME. TISERV removes all items saved in the folder (if FORCE-FLAG is set to Y) and removes the folder name from the TRANSFER name directory.
UOW Descriptions
DELETE-FOLDER

If your application requests deletion of the special folders INBOX and WASTEBASKET, these folders are emptied but the folder names remain in the system; the RETN-CODE field is set to W-CONTENTS-PURGED. The contents of WASTEBASKET can be deleted only by the logged-on correspondent.
DELETE-ITEM-REC (UOW Code 105)

DELETE-ITEM-REC deletes a data record from an item.

DEF delete-item-rec-uow.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 105.
  02 item-key.
    03 item-id.
      04 dummy PIC X(12).
    03 rec-type PIC 9(4) COMP.
    03 rec-seq-num PIC 9(4) COMP.
END.

DEF delete-item-rec-rsp.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 105.
  02 retn-code TYPE BINARY 16.
  02 retn-code-detail TYPE BINARY 16.
END.

DELETE-ITEM-REC FIELDS. The fields defined in this UOW are:

- HDR is the UOW header. The UOW-CODE value is 105.
- ITEM-KEY identifies the item to be deleted.
  ITEM-ID is the item ID of the record to be deleted.
  REC-TYPE is the type assigned to the record by your application.
  REC-SEQ-NUM is the sequence number assigned to the record.
- RETN-CODE is the return code. TISERV returns a code in this field to indicate one of the following entries:
  0  OK
  4010 E-BAD-TRANSACTION
  4035 E-ITEM-NOT-FOUND
  4041 E-ITEM-UNALTERABLE
  4047 E-REC-NOT-FOUND
UOW Descriptions
DELETE-ITEM-REC

- RETN-CODE-DETAIL is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

DELETE-ITEM-REC OPERATION. DELETE-ITEM-REC deletes the record identified by REC-TYPE and REC-SEQ-NUM from the item identified by ITEM-ID.
DELETE-MEMBER (UOW Code 221)

DELETE-MEMBER deletes one or more members from a distribution list. Members can be correspondents, distribution lists, or both.

DEF delete-member-uow.
  02 hdr.
    03 self-ident
    03 uow-code PIC AA VALUE "UW".
    TYPE BINARY 16 UNSIGNED
    VALUE 221.
  02 corr-name
  02 dlist-name
  02 num-wanted
  02 member-name
END.

DEF delete-member-rsp.
  02 hdr.
    03 self-ident
    03 uow-code PIC AA VALUE "UW".
    TYPE BINARY 16 UNSIGNED
    VALUE 221.
  02 retn-code
  02 retn-code-detail
  02 corr-name
  02 dlist-name
  02 num-returned
  02 mbr-retn-code
END.

DELETE-MEMBER FIELDS. The fields defined in this UOW are:

- HDR is the UOW header. The UOW-CODE value is 221.

- CORR-NAME is the name of the correspondent represented by the calling process. Only system administrators can enter correspondent names other than their own in this field; correspondents without system administrator privileges must enter their own names or leave the field blank. On input, you can use a partially qualified name, including wildcard characters. In the response, the fully qualified name is returned.
UOW Descriptions
DELETE-MEMBER

- **DLIST-NAME** is the name of the distribution list from which the new member is deleted. This can be a partially qualified name, including wildcard characters. In the response, the fully qualified name is returned.

- **NUM-WANTED** is the number of members to be removed from the distribution list. You can specify any number of members, restricted only by the length of the entire IPC that contains this UOW. The maximum IPC length is defined during TRANSFER system configuration.

- **MEMBER-NAME** is the name of the correspondent or distribution list to be deleted from the list identified by DLIST-NAME. This must be a fully qualified name and must exactly match a name in the distribution list, including the suffix. In the OCCURS DEPENDING ON clause, the value 5 is an arbitrary value suitable for most applications; you can reset it to any other value. You can have as many member names as specified by NUM-WANTED.

- **RETN-CODE** is the return code. TISERV returns a code in this field to indicate one of the following entries.

  To indicate successful deletion:
  
  0 OK

  To indicate problems with the correspondent name:
  
  5600 E-CORR-NSRV-ERR 5607 E-CORR-NSRV-DOWN
  5601 E-CORR-NOT-FOUND 5611 E-CORR-NET-DOWN
  5602 E-CORR-BAD-NAME 5622 E-CORR-NOT-SAME-NODE
  5604 E-CORR-NO-SUCH-NODE 5623 E-CORR-AMBIGUOUS-NAME
  5606 E-CORR-NSRV-NOT-FOUND

  To indicate problems with the distribution list name:
  
  5625 E-DLIST-NSRV-ERR 5632 E-DLIST-NSRV-DOWN
  5626 E-DLIST-NOT-FOUND 5636 E-DLIST-NET-DOWN
  5627 E-DLIST-BAD-NAME 5647 E-DLIST-NOT-SAME-NODE
  5629 E-DLIST-NO-SUCH-NODE 5648 E-DLIST-AMBIGUOUS-NAME
  5631 E-DLIST-NSRV-NOT-FOUND

  To indicate other problems:
  
  4010 E-BAD-TRANSACTION 4227 W-ERR-ON-MEMBER
  4093 E-SECURITY-VIOLATION 4902 E-ERR-PROFILE-FILE
  4201 E-CONTEXT-ERR 4914 E-ERR-DLIST-FILE

- **RETN-CODE-DETAIL** is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.
UOW Descriptions
DELETE-MEMBER

- NUM-RETURNED is the number of members deleted from the list. There will be a member return code for each member that you attempted to delete.

- MBR-RETN-CODE is a code to indicate the status of each member the UOW attempted to delete. The message OK indicates that the member was successfully deleted.

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OK</td>
</tr>
<tr>
<td>5651</td>
<td>E-MBR-NOT-FOUND</td>
</tr>
<tr>
<td>5652</td>
<td>E-MBR-BAD-NAME</td>
</tr>
<tr>
<td>5674</td>
<td>E-MBR-BAD-SUFFIX</td>
</tr>
</tbody>
</table>

DELETE-MEMBER OPERATION. The DELETE-MEMBER UOW deletes members identified by MEMBER-NAME from the distribution list identified by DLIST-NAME.
DELETE-PROFILE-REC (UOW Code 214)

DELETE-PROFILE-REC deletes a profile record from a depot.

```
DEFINE delete-profile-rec-uow.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 214.
    02 corr-name PIC X(80) VALUE SPACES.
    02 rec-type PIC 9(4) COMP.
    02 rec-seq-num PIC 9(4) COMP.
    02 depot-flag TYPE BOOLEAN VALUE "Y".
    02 filler TYPE CHARACTER 1.
END.
```

```
DEFINE delete-profile-rec-rsp.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 214.
    02 retn-code TYPE BINARY 16.
    02 retn-code-detail TYPE BINARY 16.
    02 corr-name PIC X(80).
END.
```

DELETE-PROFILE-REC FIELDS. The fields defined in this UOW are:

- HDR is the UOW header. The UOW-CODE value is 214.
- CORR-NAME is the name of the correspondent for whom the depot profile record will be deleted. This is also the correspondent represented by the calling process. Only system administrators can enter correspondent names other than their own in this field; correspondents without system administrator privileges must enter their own names or leave the field blank. On input, you can use a partially qualified name, including wildcard characters. In the response, the fully qualified name is returned.
- REC-TYPE is the type assigned to the record to be deleted, as defined by your application.
- REC-SEQ-NUM indicates the specific record to be deleted.
• DEPOT-FLAG determines whether the UOW references a depot profile record or a system control record.

   Y = a depot profile record, as indicated by CORR-NAME

   N = a system control record

• RETN-CODE is the return code. TISERV returns a code in this field to indicate one of the following entries.

   To indicate successful deletion:

   0 OK

   To indicate problems with the correspondent name:

   5600 E-CORR-NSRV-ERR
   5601 E-CORR-NOT-FOUND
   5602 E-CORR-BAD-NAME
   5604 E-CORR-NO-SUCH-NODE
   5606 E-CORR-NSRV-NOT-FOUND

   To indicate other problems:

   4010 E-BAD-TRANSACTION
   4046 E-INVALID-REC-TYPE
   4047 E-REC-NOT-FOUND
   4051 E-MUST-BE-YN
   4093 E-SECURITY-VIOLATION
   4201 E-CONTEXT-ERR
   4902 E-ERR-PROFILE-FILE

• RETN-CODE-DETAIL is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

DELETE-PROFILE-REC OPERATION. DELETE-PROFILE-REC deletes from the depot Profile file the record indicated by REC-TYPE and REC-SEQ-NUM.
DELETE-RECIP (UOW Code 115)

DELETE-RECIP deletes a recipient from a package recipient list.

```
DEF delete-recipient-uow.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 115.
  02 item-id.
    03 dummy PIC X(12).
  02 recip-name PIC X(120).
END.
```

```
DEF delete-recipient-name-rsp.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 115.
  02 retn-code TYPE BINARY 16.
  02 retn-code-detail TYPE BINARY 16.
END.
```

DELETE-RECIP FIELDS. The fields defined in this UOW are:

- **HDR** is the UOW header. The UOW-CODE value is 115.

- **ITEM-ID** identifies the package from whose recipient list the recipient is to be deleted. This is the item ID of the header for that package.

- **RECIP-NAME** is the name of the recipient to be deleted. This field must match exactly the recipient name as stored in the list.

If the name was added by the ADD-RECIP UOW with deferred resolution, the stored name can contain wildcard characters; in this case, you must specify exactly that pattern, including wildcard characters. Wildcard characters in RECIP-NAME are not expanded; instead, they are treated as any other characters in determining whether RECIP-NAME matched a name in the list.
• RETN-CODE is the return code. TISERV returns a code in this field to indicate one of the following entries:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OK</td>
</tr>
<tr>
<td>4010</td>
<td>E-BAD-_TRANSACTION</td>
</tr>
<tr>
<td>4035</td>
<td>E-ITEM-NOT-FOUND</td>
</tr>
<tr>
<td>4041</td>
<td>E-ITEM-UNALTERABLE</td>
</tr>
<tr>
<td>4042</td>
<td>E-ITEM-NOT-PKG-HDR</td>
</tr>
<tr>
<td>5751</td>
<td>E-RECIP-NOT-FOUND</td>
</tr>
<tr>
<td>5752</td>
<td>E-RECIP-BAD-NAME</td>
</tr>
<tr>
<td>5774</td>
<td>E-RECIP-BAD-SUFFIX</td>
</tr>
</tbody>
</table>

• RETN-CODE-DETAIL is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

DELETE-RECIP OPERATION. DELETE-RECIP deletes the name identified by RECIP-NAME from the recipient list for the package identified by ITEM-ID.
DETACH-COMPNT

DETACH-COMPNT (UOW Code 113)

DETACH-COMPNT detaches a component item from a parent.

DEF detach-compnt-uow.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 113.
  02 parent-item-id.
    03 dummy PIC X(12).
    02 rel-position TYPE BINARY 16.
    02 compnt-id.
    03 dummy PIC X(12).
END.

DEF detach-compnt-rsp.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 113.
    02 retn-code TYPE BINARY 16.
    02 retn-code-detail TYPE BINARY 16.
    02 detached-position TYPE BINARY 16.
    02 detached-compnt.
    03 dummy PIC X(12).
END.

DETACH-COMPNT FIELDS. The fields defined in this UOW are:

- HDR is the UOW header. The UOW-CODE value is 113.
- PARENT-ITEM-ID identifies the parent item.
- REL-POSITION is the position that the component occupies in the component list. This record can also be addressed through the COMPNT-ID field, but REL-POSITION provides faster access. If you use the COMPNT-ID field you must set the REL-POSITION field to zero.
- COMPNT-ID is the ID of the component item, but provides slower access to the record than a reference through REL-POSITION. You can use COMPNT-ID only if you set REL-POSITION to zero.
• RETN-CODE is the return code. TISERV returns a code in this field to indicate one of the following entries:

```
<table>
<thead>
<tr>
<th>RETN-CODE</th>
<th>DETAIL</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OK</td>
</tr>
<tr>
<td>4010</td>
<td>E-BAD-TRANSACTION</td>
</tr>
<tr>
<td>4037</td>
<td>E-PARENT-NOT-FOUND</td>
</tr>
<tr>
<td>4039</td>
<td>E-COMPNT-NOT-FOUND</td>
</tr>
<tr>
<td>4040</td>
<td>E-BAD-ITEM-DESCR</td>
</tr>
<tr>
<td>4041</td>
<td>E-ITEM-UNALTERABLE</td>
</tr>
<tr>
<td>4057</td>
<td>E-INVALID-REL-POSITION</td>
</tr>
</tbody>
</table>
```

• RETN-CODE-DETAIL is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

• DETACHED-POSITION is the position previously occupied in the components list by the detached component.

• DETACHED-COMPNT is the item ID of the detached component.

**DETACH-COMPNT OPERATION.** DETACH-COMPNT detaches a component item from its parent item by removing it from the parent components list. You can reference this record in the UOW either by its relative position within the components list (with the REL-POSITION field) or by its item ID (with the COMPNT-ID field).

If this deletion is successful, the component count in the parent item descriptor and the parent count in the component item descriptor are both decremented by 1; the relative position of all succeeding items in the components list is also decremented by 1. If component 2 is detached, for example, component 3 becomes component 2.
END-SESSION

END-SESSION (UOW Code 102)

END-SESSION terminates a session.

```lisp
DEF end-session-uow.
  02 hdr.
    03 self-ident
    03 uow-code PIC AA VALUE "UW".
    TYPE BINARY 16 UNSIGNED VALUE 102.
  END.

DEF end-session-rsp.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 102.
    02 retn-code TYPE BINARY 16.
    02 retn-code-detail TYPE BINARY 16.
  END.
```

END-SESSION FIELDS. The fields defined in this UOW are:

- **HDR** is the UOW header. The UOW-CODE value is 102.
- **RETN-CODE** is the return code. TISERV returns a code in this field to indicate one of the following entries:
  
  0  OK
  4010  E-BAD-TRANSACTION

- **RETN-CODE-DETAIL** is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

END-SESSION OPERATION. END-SESSION terminates the current session and removes any items residing in the WASTEBASKET folder and in the internal temporary folder.

A successful execution sets the session ID in the returned IPC header to zero. This sets up the IPC header automatically for a new START-SESSION UOW.
GET-AGENT-SELECT (UOW Code 207)

GET-AGENT-SELECT retrieves agent selection criteria for a depot.

```
DEF get-agent-select-uow.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 207.
  02 corr-name PIC X(80) VALUE SPACES.
  02 agent-rec-num TYPE BINARY 16 UNSIGNED.
  02 action TYPE CHARACTER 1.
  02 filler TYPE CHARACTER 1.
END.
```

```
DEF get-agent-select-rsp.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 207.
  02 retn-code TYPE BINARY 16.
  02 retn-code-detail TYPE BINARY 16.
  02 corr-name PIC X(80).
  02 action TYPE CHARACTER 1.
  02 filler TYPE CHARACTER 1.
  02 agent-rec-num TYPE BINARY 16 UNSIGNED.
  02 agent-name TYPE CHARACTER 80.
  02 agent-type TYPE CHARACTER 1.
  02 agent-flag TYPE CHARACTER 1.
  02 applic-id-for-logon PIC 9(4) COMP.
  02 applic-id-low PIC 9(4) COMP.
  02 applic-id-high PIC 9(4) COMP.
  02 agent-sel-low PIC 9(4) COMP.
  02 agent-sel-high PIC 9(4) COMP.
  02 agent-data TYPE CHARACTER 80.
END.
```

GET-AGENT-SELECT FIELDS. The fields defined in this UOW are:

- **HDR** is the UOW header. The UOW-CODE value is 207.
UOW Descriptions
GET-AGENT-SELECT

- CORR-NAME is the name of the correspondent represented by the calling process. Only system administrators can enter correspondent names other than their own in this field; correspondents without system administrator privileges must enter their own names or leave the field blank. On input, you can use a partially qualified name, including wildcard characters. In the response, the fully qualified name is returned.

- AGENT-REC-NUM is the position of the record that defines the agent in the agent list for the depot. This field is used in conjunction with the ACTION field. In the response, the number of the record accessed is returned.

- ACTION specifies which record should be read, relative to the value supplied in AGENT-REC-NUM.
  
  E = read the record denoted by AGENT-REC-NUM

  A = read the record following the record denoted by AGENT-REC-NUM

If the record indicated by AGENT-REC-NUM does not exist, the warning W-EOF is returned in the RETN-CODE field.

Typically, an application reads selection criteria for several agents.

To read the criteria for the first agent (record) in the agent list, set

AGENT-REC-NUM = 0

ACTION = A

To read the criteria for the next agent, set

AGENT-REC-NUM = the value returned on the previous call

ACTION = A

In the response, no data is returned in this field. The field is included only to simplify the use of the GET-AGENT-SELECT UOW response as an ALTER-AGENT-SELECT request.

- RETN-CODE is the return code. TISERV returns a code in this field to indicate one of the following entries.

  To indicate successful retrieval:

  0 OK
To indicate problems with the correspondent name:

5600 E-CORR-NSRV-ERR
5601 E-CORR-NOT-FOUND
5602 E-CORR-BAD-NAME
5604 E-CORR-NO-SUCH-NODE
5606 E-CORR-NSRV-NOT-FOUND

5607 E-CORR-NSRV-DOWN
5611 E-CORR-NET-DOWN
5622 E-CORR-NOT-SAME-NODE
5623 E-CORR-AMBIGUOUS-NAME

To indicate other problems:

4001 W-EOF
4010 E-BAD-TRANSACTION
4201 E-CONTEXT-ERR
4229 E-MUST-BE-EA
4293 E-SECURITY-VIOLATION
4902 E-ERR-PROFILE-FILE

• RETN-CODE-DETAIL is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

• AGENT-NAME is the name of the SCREEN COBOL program or server class that operates as the agent. You determine whether this name identifies a SCREEN COBOL program or a server class with the AGENT-TYPE field.

• AGENT-TYPE determines whether the program denoted by AGENT-NAME is a SCREEN COBOL program or a server class name.

  Y = a SCREEN COBOL program
  N = a server class name

• AGENT-FLAG determines whether the agent requires the associated TAREQ to begin a session on its behalf before the agent can run.

  Y = session initiation is required
  N = session initiation is not required

• APPLIC-ID-FOR-LOGON is the application ID to be used in establishing the session when the AGENT-FLAG is set to Y.

• APPLIC-ID-LOW, APPLIC-ID-HIGH, AGENT-SEL-LOW, and AGENT-SEL-HIGH define numeric ranges used to select the agent. These fields are used as noted in the discussion of the ALTER-AGENT-SELECT UOW.

• AGENT-DATA is data that your application passes to the agent; its use is defined by the agent itself.
NOTE

The default agent configuration screen allows for viewing and modifying only the first 78 characters of the 80-character AGENT-DATA field. If you are providing an agent that requires 79 or 80 characters of agent data, you must also provide a SCREEN COBOL program for configuring such agents.

GET-AGENT-SELECT OPERATION. GET-AGENT-SELECT retrieves the requested agent selection entry from the depot Profile file. The agent is identified by AGENT-REC-NUM. The list of agents for the depot is arranged by agent record number. Existing agents can have AGENT-REC-NUM values from 1 through 500. A value of zero can be used for the read-after operation specified by A in the ACTION field. A read operation returns the AGENT-REC-NUM of the agent just read.

To read the list of agents in order, set AGENT-REC-NUM to zero and then perform read-after operations, using the most recently returned AGENT-REC-NUM until W-EOF occurs in the RETN-CODE field.

For further information about agent selection criteria, see the discussion of the ALTER-AGENT-SELECT UOW.
GET-CONFIG-NAME (UOW Code 132)

GET-CONFIG-NAME retrieves the configured name for a TRANSFER file, process, or other special entity.

DEF get-config-name-uow.
  02 hdr.
  03 self-ident PIC AA VALUE "UW".
  03 uow-code TYPE BINARY 16 UNSIGNED VALUE 132.
  02 network-node-name PIC X(32).
  02 entity-name PIC X(32).
END.

DEF get-config-name-rsp.
  02 hdr.
  03 self-ident PIC AA VALUE "UW".
  03 uow-code TYPE BINARY 16 UNSIGNED VALUE 132.
  02 retn-code TYPE BINARY 16.
  02 retn-code-detail TYPE BINARY 16.
  02 configured-name PIC X(80).
END.

GET-CONFIG-NAME FIELDS. The fields defined in this UOW are:

- **HDR** is the UOW header. The UOW-CODE value is 132.

- **NETWORK-NODE-NAME** is the name of the network node at which the TRANSFER entity is defined. This node name should include the \ at the beginning of the name. To indicate the local node, you can enter blanks. This is a TRANSFER simple name.

- **ENTITY-NAME** is the name of the TRANSFER entity for which your application requests the fully qualified name configured in the system. This name is defined in the TRANSFER name configuration directory. The following entries in this field are meaningful:
  
  DEFAULT-MODEL-DEPOT
  PATHMON-PROCESS
  TISERV-SERVER-CLASS
  INBOX-FOLDER
  WASTEBASKET-FOLDER
Other entries, though legal, should not be used in this field. They are intended for internal use only by Tandem analysts and support personnel.

- **RETN-CODE** is the return code. TISERV returns a code in this field to indicate one of the following:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OK</td>
</tr>
<tr>
<td>4201</td>
<td>E-CONTEXT-ERR</td>
</tr>
<tr>
<td>5700</td>
<td>E-ALIAS-NSRV-ERR</td>
</tr>
<tr>
<td>5701</td>
<td>E-ALIAS-NOT-FOUND</td>
</tr>
<tr>
<td>5702</td>
<td>E-ALIAS-BAD-NAME</td>
</tr>
<tr>
<td>5703</td>
<td>E-ALIAS-BAD-TYPE</td>
</tr>
<tr>
<td>5704</td>
<td>E-ALIAS-NO-SUCH-NODE</td>
</tr>
<tr>
<td>5705</td>
<td>E-ALIAS-SECURITY</td>
</tr>
<tr>
<td>5706</td>
<td>E-ALIAS-NSRV-NOT-FOUND</td>
</tr>
<tr>
<td>5707</td>
<td>E-ALIAS-NSRV-DOWN</td>
</tr>
<tr>
<td>5711</td>
<td>E-ALIAS-NET-DOWN</td>
</tr>
</tbody>
</table>

- **RETN-CODE-DETAIL** is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

- **CONFIGURED-NAME** is the fully qualified configured name of the requested entity.

**GET-CONFIG-NAME OPERATION.** GET-CONFIG-NAME retrieves the configured name for the entity identified by ENTITY-NAME. This name, which is stored in the TRANSFER name configuration directory, is returned in the CONFIGURED-NAME field.

NOTE

This UOW does not require establishment of a session for its execution.

Names in the name configuration directory, which is part of the TRANSFER name directory, must correspond to real entities in order for TRANSFER to operate properly. This directory contains the names of all entities required by TRANSFER in its running environment. If names do not correspond, problems arise, some of which are not immediately obvious. For example, if PATHMON is running with a process name other than the one specified in the name configuration directory, your application might not discover this until it tries (and fails) to transmit packages across the network. These names are established during the initialization of TRANSFER through the XBEGIN and NLOAD modules, as described in the TRANSFER Delivery System Management and Administration Guide.
The configured name for TISERV-SERVER-CLASS is usually "TISERV". If you are using a server class name other than TISERV, you must still have a TISERV server class defined. The application can then perform the following:

1. At initialization time, send a GET-CONFIG-NAME request to the server class named "TISERV", asking for the name of the TISERV-SERVER-CLASS.

2. Send all future TISERV requests to the server class whose name was returned in step 1.
GET-FOLDER-ORDER retrieves the ordering criteria for a folder.

**GET-FOLDER-ORDER FIELDS.** The fields defined in this UOW are:

- **HDR** is the UOW header. The UOW-CODE value is 233.
• CORR-NAME is the name of the correspondent represented by the calling process. Only system administrators can enter correspondent names other than their own in this field; correspondents without system administrator privileges must enter their own names or leave the field blank. On input, you can use a partially qualified name, including wildcard characters. In the response, the fully qualified correspondent name is returned.

• FOLDER-NAME is the name of the folder to be retrieved. The name must be either a TRANSFER simple name or a fully qualified name.

• RETN-CODE is the return code. TISERV returns a code in this field to indicate one of the following entries.

To indicate successful retrieval of the folder:

0 OK

To indicate problems with the correspondent name:

5600 E-CORR-NSRV-ERR 5607 E-CORR-NSRV-DOWN
5601 E-CORR-NOT-FOUND 5611 E-CORR-NET-DOWN
5602 E-CORR-BAD-NAME 5622 E-CORR-NOT-SAME-NODE
5604 E-CORR-NO-SUCH-NODE 5623 E-CORR-AMBIGUOUS-NAME
5606 E-CORR-NSRV-NOT-FOUND

To indicate problems with the folder name:

5675 E-FLD-NSRV-ERR 5682 E-FLD-NSRV-DOWN
5676 E-FLD-NOT-FOUND 5686 E-FLD-NET-DOWN
5677 E-FLD-BAD-NAME 5697 E-FLD-NOT-SAME-NODE
5679 E-FLD-NO-SUCH-NODE 5698 E-FLD-AMBIGUOUS-NAME
5681 E-FLD-NSRV-NOT-FOUND

To indicate other problems:

4010 E-BAD-TRANSACTION 4201 E-CONTEXT-ERR
4093 E-SECURITY-VIOLATION 4902 E-ERR-PROFILE-FILE
4105 E-CONCURRENT-FLD-UPDATE
4106 E-BAD-ORD-CRITERIA

• RETN-CODE-DETAIL is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

• ORDERING-DISCIPLINE specifies what information TISERV uses as the ordering key when saving items in the folder as follows.
### UOW Descriptions

#### GET-FOLDER-ORDER

<table>
<thead>
<tr>
<th>Entry</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIME-SAVED (T)</td>
<td>Items are saved in chronological order of the time a SAVE-ITEM UOW is executed for the items.</td>
</tr>
<tr>
<td>CREATOR-NAME (C)</td>
<td>Items are saved in alphabetic order by creator name.</td>
</tr>
<tr>
<td>EARLIEST-DELIV-DATE (E)</td>
<td>Items that are package headers are saved by earliest delivery date; items that are not package headers are saved by earliest creation date.</td>
</tr>
<tr>
<td>APPLIC-DEFINED (A)</td>
<td>Items are saved in order of the ordering key specified by the application in the SAVE-ITEM-BY-KEY UOW.</td>
</tr>
</tbody>
</table>

- **APPLIC-ORDER-TYPE** is meaningful only if ORDERING-DISCIPLINE is APPLIC-DEFINED (A).

This field is not interpreted by TRANSFER, but is stored as part of the folder's ordering criteria. This field enables a TRANSFER application to have several types of APPLIC-DEFINED ordering; the APPLIC-ORDER-TYPE can be used to distinguish between types.

This field is set to 0 if the ORDERING-DISCIPLINE is not APPLIC-DEFINED.

- **OPTIONS** are the additional criteria for saving items. Two options are provided.

  **ASCENDING-SEQUENCE** indicates whether items are saved within the folder in ascending or descending key order.

  - Y = Items are saved in ascending key order.
  - N = Items are saved in descending key order.

  **ALLOW-DUPLICATES** indicates whether or not duplicate keys are allowed.

  - Y = Duplicate ordering keys are allowed.
  - N = Duplicate ordering keys are not allowed. This field has no effect if the ORDERING-DISCIPLINE is TIME-SAVED (T); TIME-SAVED ordering already ensures a unique key.
Reserved-2 through Reserved-7 are reserved for use by Tandem; these fields are always set to N.

**GET-FOLDER-ORDER Operation.** GET-FOLDER-ORDER retrieves the ordering criteria for the name assigned in FOLDER-NAME.
GET-ITEM-COMPNT-AOl retrieves one or more component items in a parent item.

**GET-ITEM-COMPNT-AOl**

**UOW Descriptions**

**GET-ITEM-COMPNT-AOl** (UOW Code 137)

**GET-ITEM-COMPNT-AOl FIELDS.** The fields defined in this UOW are:

- **HDR** is the UOW header. The UOW-CODE value is 137.

```plaintext
DEF get-item-compnt-aOl-uow.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 137.

  02 item-id.
    03 dummy PIC X(12).
    02 start-position TYPE BINARY 16.

  02 options.
    03 filter-by-item-type TYPE BOOLEAN.
    03 filter-by-compnt-type TYPE BOOLEAN.

  02 filter-values.
    03 item-type PIC 9(4) COMP.
    03 compnt-type PIC 9(4) COMP.

  02 num-requested TYPE BINARY 16 UNSIGNED VALUE 10.
END.

DEF get-item-compnt-aOl-rsp.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 137.

    02 retn-code TYPE BINARY 16.
    02 retn-code-detail TYPE BINARY 16.
    02 num-returned TYPE BINARY 16 UNSIGNED.
    02 compnt-entry OCCURS 0 TO !num-requested! 10 TIMES DEPENDING ON num-returned.

    03 item-id.
      04 dummy PIC X(12).
    03 rel-position TYPE BINARY 16.
    03 item-type PIC 9(4) COMP.
    03 compnt-type PIC 9(4) COMP.
END.
```
ITEM-ID identifies the parent item to which the components belong.

START-POSITION is the relative position in the components list of the first item for which data is to be returned.

OPTIONS provides criteria for selecting component items by ITEM-TYPE or by COMPNT-TYPE. ITEM-TYPE is a numeric value that the application defines and then assigns to the item through the CREATE-ITEM UOW. COMPNT-TYPE is a numeric value that TRANSFER saves with the component when the component item is attached to the parent through the ATTACH-COMPNT-AOl UOW.

In these options, the entry Y selects the option, while the entry N suppresses the option.

- To select filtering by item type:
  FILTER-BY-ITEM-TYPE = Y
  ITEM-TYPE = the item type to be retrieved

- To select filtering by component type:
  FILTER-BY-COMPNT-TYPE = Y
  COMPNT-TYPE = the component type to be retrieved

NUM-REQUESTED is the total number of component items for which data is to be returned, starting with the item indicated by START-POSITION. As an example, to return 10 components beginning with the component at relative position 6, you would set START-POSITION to 6 and NUM-REQUESTED to 10. The NUM-REQUESTED field can contain values ranging from 1 through 400; the value in this field directly affects the size of the response.

RETN-CODE is the return code. TISERV returns a code in this field to indicate one of the following entries:

0  OK
4001 W-EOF
4035 E-ITEM-NOT-FOUND
4040 E-BAD-ITEM-DESCR
4051 E-MUST-BE-YN
4057 E-INVALID-REL-POSITION
4092 E-INVALID-NUM-RQSTD

RETN-CODE-DETAIL is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

NUM-RETURNED is the number of components for which TISERV returned data in the response.
UOW Descriptions
GET-ITEM-COMPNT-A01

- COMPNT-ENTRY contains the item ID (in the ITEM-ID field), the relative position (in the REL-POSITION field), the item type (in the ITEM-TYPE field), and the component type (in the COMPNT-TYPE field) of each component, as returned by TISERV. The size of this array must be declared consistently with the length of the data requested by START-POSITION and NUM-REQUESTED.

GET-ITEM-COMPNT-A01 OPERATION. GET-ITEM-COMPNT-A01 returns, in COMPNT-ENTRY, the item ID and item type of each component item requested by ITEM-ID, START-POSITION, and NUM-REQUESTED. TRANSFER lets you access the associated component items directly at any time during the current session.

If filtering by item type or component type is requested, any component whose item type or component type does not match the specified filter values is ignored and is not returned.

If TISERV reaches the end of the components list before retrieving all specified components, the number of entries returned is less than the total number requested. In this case, the warning W-EOF is returned in the RETN-CODE field.
GET-ITEM-DESCR (UOW Code 122)

GET-ITEM-DESCR retrieves fields from an item descriptor.

DEF get-item-descr-uow.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 122.
  02 item-id.
    03 dummy PIC X(12).
END.

DEF get-item-descr-rsp.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 122.
  02 retn-code TYPE BINARY 16.
  02 retn-code-detail TYPE BINARY 16.
  02 update-control PIC S9(4) COMP.
  02 item-type PIC 9(4) COMP.
  02 parent-count TYPE BINARY 16 UNSIGNED.
  02 compnt-count TYPE BINARY 16 UNSIGNED.
  02 item-descr-flags.
    03 is-pkg-hdr TYPE BOOLEAN.
    03 byte REDEFINES IS-PKG-HDR PIC X.
    03 unalterable TYPE BOOLEAN.
    03 submitted TYPE BOOLEAN.
    03 cancelled TYPE BOOLEAN.
    03 cloned TYPE BOOLEAN.
    03 rel-date-earliest TYPE BOOLEAN.
    03 rel-date-latest TYPE BOOLEAN.
    03 rel-date-expiration TYPE BOOLEAN.
    03 reserved-8 TYPE BOOLEAN VALUE "N".
    03 reserved-9 TYPE BOOLEAN VALUE "N".
    03 reserved-10 TYPE BOOLEAN VALUE "N".
    03 reserved-11 TYPE BOOLEAN VALUE "N".
    03 reserved-12 TYPE BOOLEAN VALUE "N".
    03 reserved-13 TYPE BOOLEAN VALUE "N".
    03 reserved-14 TYPE BOOLEAN VALUE "N".
    03 reserved-15 TYPE BOOLEAN VALUE "N".
  02 creation-date.
    03 year PIC 9(4).
    03 month PIC 9(2).
    03 day-of-month PIC 9(2).
    03 hour PIC 9(2).
03 minute          PIC 9(2).
03 second         PIC 9(2).
02 creator-name   PIC X(120).
02 submitted-date.
  03 year           PIC 9(4).
  03 month          PIC 9(2).
  03 day-of-month   PIC 9(2).
  03 hour           PIC 9(2).
  03 minute         PIC 9(2).
  03 second         PIC 9(2).
02 earliest-deliv-date.
  03 year           PIC 9(4).
  03 month          PIC 9(2).
  03 day-of-month   PIC 9(2).
  03 hour           PIC 9(2).
  03 minute         PIC 9(2).
  03 second         PIC 9(2).
  03 delta-time     REDEFINES DATE-TIME.
  04 quantity       PIC 9(4) COMP.
  04 units          PIC A.
  04 filler         PIC X.
02 latest-deliv-date.
  03 year           PIC 9(4).
  03 month          PIC 9(2).
  03 day-of-month   PIC 9(2).
  03 hour           PIC 9(2).
  03 minute         PIC 9(2).
  03 second         PIC 9(2).
  03 delta-time     REDEFINES DATE-TIME.
  04 quantity       PIC 9(4) COMP.
  04 units          PIC A.
  04 filler         PIC X.
02 expiration-date.
  03 year           PIC 9(4).
  04 month          PIC 9(2).
  04 day-of-month   PIC 9(2).
  04 hour           PIC 9(2).
  04 minute         PIC 9(2).
  04 second         PIC 9(2).
  03 delta-time     REDEFINES DATE-TIME.
  04 quantity       PIC 9(4) COMP.
  04 units          PIC A.
  04 filler         PIC X.
02 sender-zone-offset  PIC S9(4) COMP.
02 priority       PIC 9(3) COMP.
GET-ITEM-DESCR FIELDS. The fields defined in this UOW are:

- **HDR** is the UOW header. The UOW-CODE value is 122.
- **ITEM-ID** identifies the item whose descriptor record is to be retrieved.
- **RETN-CODE** is the return code. TISERV returns a code in this field to indicate one of the following entries:
  
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OK</td>
</tr>
<tr>
<td>4035</td>
<td>E-ITEM-NOT-FOUND</td>
</tr>
</tbody>
</table>

- **RETN-CODE-DETAIL** is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.
- **UPDATE-CONTROL** is the current update control count. This field will contain the value 1 if the descriptor has never been altered.
- **ITEM-TYPE** is the type of item, as defined by the application.
UOW Descriptions
GET-ITEM-DESCR

- PARENT-COUNT is the number of parents that identify this item as a component.

- COMPNT-COUNT is the number of component items listed in the components list for this item.

- ITEM-DESCR-FLAGS describe miscellaneous flags that appear in the item descriptor.

The fields within ITEM-DESCR-FLAGS are as follows:

IS-PKG-HDR indicates whether the item is a package header.

UNALTERABLE indicates whether the item is unalterable. If this field is set to Y, the entire item is unalterable; and the item descriptor, components list, and data records cannot be changed.

SUBMITTED indicates whether the package has been posted.

CANCELLED indicates whether the package, if posted, was canceled.

CLONED indicates whether the package is a copy of a package that was transported from another node.

REL-DATA-EARLIEST, REL-DATA-LATEST, and REL-DATA-EXPIRATION determine whether the entries specified in the EARLIEST-DELIV-DATE, LATEST-DELIV-DATE, and EXPIRATION-DATE fields are relative dates with respect to particular delivery milestones, or absolute calendar dates.

Y = relative dates

N = absolute dates

RESERVED-8 through RESERVED-15 are reserved for use by Tandem.

- CREATION-DATE is the date that the item was created.

- CREATOR-NAME is the name of the correspondent who created the item.

NOTE

If the item is not a package header, the following fields will contain binary zeros.

- SUBMITTED-DATE is the date and time at which the package was posted. If the package has never been submitted, this field contains zeros.
• EARLIEST-DELIV-DATE is the earliest date at which the package can be delivered, specified as either an absolute calendar date or as a number of time units from submission time. An absolute or relative date is indicated by N or Y, respectively, in the REL-DATE-EARLIEST field, as described under ITEM-DESCR-FLAGS.

NOTE

When a package is submitted, all relative dates in the EARLIEST-DELIV-DATE field and the LATEST-DELIV-DATE and EXPIRATION-DATE fields are automatically changed to absolute calendar dates.

• LATEST-DELIV-DATE is the latest date at which TRANSFER can deliver the package, specified either as an absolute calendar date or as a number of time units from EARLIEST-DELIV-DATE.

• EXPIRATION-DATE is the date and time at which the package expires, specified either as an absolute calendar date or as a number of time units from LATEST-DELIV-DATE. If expiration has not been requested, this field contains zeros.

• SENDER-ZONE-OFFSET is the difference, in minutes, between the time at the sender node and the time local to the current session.

• PRIORITY is the package priority. The field can have a value from 0 (lowest priority) to 199 (highest priority).

• APPLIC-ID is the numeric application ID associated with the session during which the item was created. This value is used as agent selection criteria at recipient depots. Refer to the START-SESSION and ALTER-AGENT-SELECT UOWs.

• AGENT-SEPARATOR is the agent selector criteria, as created by the ALTER-AGENT-SELECT UOW. This is a number that is stored with the package to determine which agents are invoked when the package is delivered. Refer to the ALTER-AGENT-SELECT UOW.

• DELIV-CONTROL-FLAGS specifies whether or not the package is certified. Refer to the ALTER-ITEM-DESCR UOW.

• ERR-PKG-SUPPRESS-FLAGS specifies whether or not error reporting is suppressed. Refer to the ALTER-ITEM-DESCR UOW.

GET-ITEM-DESCR OPERATION. GET-ITEM-DESCR returns the item descriptor fields for the item identified by ITEM-ID.
UOW Descriptions
GET-ITEM-REC

GET-ITEM-REC (UOW Code 125)

GET-ITEM-REC retrieves one or more data records from an item.

DEF get-item-rec-uow.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 125.
  02 starting-key.
    03 item-id PIC X(12).
    04 dummy
    03 rec-type PIC S9(4) COMP.
    03 rec-seq-num PIC S9(4) COMP.
  02 options.
    03 skip-exact TYPE BOOLEAN.
    03 any-rec-type TYPE BOOLEAN.
    03 any-seq-num TYPE BOOLEAN.
    03 reserved-3 TYPE BOOLEAN VALUE "N".
  02 num-requested TYPE BINARY 16 UNSIGNED VALUE 20.
  02 max-datasize TYPE BINARY 16 UNSIGNED.
  02 pad-char PIC X.
  02 filler PIC X.
END.

DEF get-item-rec-rsp.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 125.
  02 retn-code TYPE BINARY 16.
  02 retn-code-detail TYPE BINARY 16.
  02 num-returned TYPE BINARY 16 UNSIGNED.
  02 recs-returned OCCURS 0 TO !num-requested! 20 TIMES DEPENDING ON num-returned.
    03 rec-type PIC 9(4) COMP.
    03 rec-seq-num PIC 9(4) COMP.
    03 data-len TYPE BINARY 16 UNSIGNED.
    03 data-string PIC !X(max-datasize)! X(80).
END.
GET-ITEM-REC FIELDS. The fields defined in this UOW are:

- **HDR** is the UOW header. The UOW-CODE value is 125.

- **STARTING-KEY** identifies the first possible item data record that can be retrieved. The record is determined by the ITEM-ID, REC-TYPE, and REC-SEQ-NUM specified.

- **OPTIONS** provides criteria for selecting the item data records.

  **SKIP-EXACT** determines whether TISERV begins its retrieval with the record specified, or the record after the one specified.

  - **Y** = Begin retrieval with the record after the record specified.
  - **N** = Begin retrieval with the record specified.

  If the beginning record (specified by STARTING-KEY) is not present in the file, the record after the one specified is retrieved regardless of the setting of SKIP-EXACT.

  **ANY-REC-TYPE** determines whether records not having the record type specified in STARTING-KEY also can be returned.

  - **Y** = Records of record type greater than REC-TYPE also can be returned. ANY-SEQ-NUM is ignored; TRANSFER retrieves up to NUM-REQUESTED records of type REC-TYPE with a record sequence number greater than or equal to REC-SEQ-NUM or with type greater than REC-TYPE and any sequence number.
  - **N** = Only records of type REC-TYPE will be returned.

  **ANY-SEQ-NUM** is meaningful only if **ANY-REC-TYPE** is set to **N**.

    - **ANY-REC-TYPE = N and ANY-SEQ-NUM = N**
      
      One record at most will be returned; the record will match exactly the REC-TYPE and REC-SEQ-NUM specified in STARTING-KEY provided the SKIP-EXACT field is set to N.

    - **ANY-REC-TYPE = N and ANY-SEQ-NUM = Y**
      
      Up to NUM-REQUESTED records of type REC-TYPE with sequence numbers greater than or equal to REC-SEQ-NUM will be returned.

  **RESERVED-3** is reserved for use by Tandem; this field must be set to N.
NUM-REQUESTED specifies the number of records to be returned. This value directly affects the length of the response.

The records are returned starting with the lowest record type and lowest sequence number matching the STARTING-KEY and OPTIONS selected. All records within a record type are returned in sequence number order before records of the next higher record type.

For example, if GET-ITEM-REC with

- REC-TYPE = 2
- REC-SEQ-NUM = 0
- SKIP-EXACT = N
- ANY-REC-TYPE = N
- ANY-SEQ-NUM = Y
- NUM-REQUESTED = 6

were specified, the records returned might be as illustrated in Figure 5-4.

![Figure 5-4. Record Retrieval by GET-ITEM-REC](image)

MAX-DATASIZE refers to the data portion of each record; this value is the maximum number of bytes that can be retrieved. Records that are shorter than this length are automatically padded with the padding character specified by PAD-CHAR. Records that are longer than this length are truncated, but the length returned by DATA-LEN is the actual length prior to truncation.
You should specify an even-numbered maximum length so that all elements of the returned array are word aligned; but if you specify an odd-numbered maximum length, TRANSFER automatically increments this value by 1. MAX-DATASIZE, like NUM-REQUESTED, directly affects the length of the response.

- **PAD-CHAR** is the character used to pad returned records that are shorter than the maximum length specified by MAX-DATASIZE.

- **RETN-CODE** is the return code. TISERV returns a code in this field to indicate one of the following entries.

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OK</td>
</tr>
<tr>
<td>4001</td>
<td>W-EOF</td>
</tr>
<tr>
<td>4035</td>
<td>E-ITEM-NOT-FOUND</td>
</tr>
<tr>
<td>4051</td>
<td>E-MUST-BE-YN</td>
</tr>
<tr>
<td>4052</td>
<td>E-RESERVED-MUST-BE-N</td>
</tr>
<tr>
<td>4086</td>
<td>W-DATASIZE-ADJUSTED</td>
</tr>
<tr>
<td>4087</td>
<td>E-INVALID-MAX-DATASIZE</td>
</tr>
<tr>
<td>4089</td>
<td>W-DATA-TRUNCATED</td>
</tr>
<tr>
<td>4092</td>
<td>E-INVALID-NUM-RQSTD</td>
</tr>
</tbody>
</table>

- **RETN-CODE-DETAIL** is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

- **NUM-RETURNED** is the number of records returned by TRANSFER in the response.

- **RECS-RETURNED** is the data array that contains the records retrieved. For this array, you must define limits and element sizes that are consistent with the records requested in the UOW. Within this structure:

  REC-TYPE contains the record type.

  REC-SEQ-NUM contains the record sequence number.

  DATA-LEN contains the length of the data (non-key) portion of the record prior to retrieval, regardless of the value specified by MAX-DATASIZE.

  DATA-STRING contains the data portion of the record. The length of the data in DATA-STRING can be modified, but should be consistent with the value in MAX-DATASIZE.

**GET-ITEM-REC OPERATION.** GET-ITEM-REC retrieves one or more data records from the item identified by STARTING-KEY, and places them in the array named RECS-RETURNED. The values supplied in OPTIONS and NUM-REQUESTED determine which records are retrieved. Because of SCREEN COBOL restrictions that prohibit multivarying arrays, you must deal with records of fixed length, as specified in MAX-DATASIZE.
UOW Descriptions
GET-ITEM-REC

The combination of OPTIONS selected determines which portion of the record key (consisting of item ID, record type, and sequence number) is used in selecting the record, as illustrated in Figure 5-5. Regardless of the OPTIONS selected, however, TRANSFER never retrieves a record whose key is less than STARTING-KEY considered in its entirety.

![Figure 5-5. Record Key Use](image)

If there are fewer records than specified by NUM-REQUESTED, the number of records retrieved is less than the number requested. In this case, the warning W-EOF is returned in the RETN-CODE field.

TRANSFER does not retrieve records for different items in a single UOW—in other words, retrievals cannot cross item boundaries.
GET-NEXT-SESSION (UOW Code 200)

GET-NEXT-SESSION returns names of correspondents with active sessions. This UOW can only be issued by correspondents with system administrator privileges.

```
DEF get-next-session-uow.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 200.
  02 filler
  02 start-session-id.
    03 dummy PIC X(18).
  02 num-wanted
  02 filler
END.
```

```
DEF get-next-session-rsp.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 200.
  02 retn-code
  02 retn-code-detail
  02 start-session-id.
    03 dummy PIC X(18).
  02 num-wanted
  02 num-returned TYPE BINARY 16 UNSIGNED VALUE 0.
  02 session-data OCCURS 0 TO 15 TIMES DEPENDING ON num-returned.
    03 corr-name PIC X(80).
    03 session-id.
    04 dummy PIC X(18).
END.
```

**GET-NEXT-SESSION FIELDS.** The fields defined in this UOW are:

- HDR is the UOW header. The UOW-CODE value is 200.
UOW Descriptions
GET-NEXT-SESSION

- START-SESSION-ID selects the session for which information is to be returned by TISERV. On the first call, you should set START-SESSION-ID to zero to read the information for the first session, as recorded in the Session file. On successive calls, the START-SESSION-ID returned in the response should be specified to retrieve the information for the next session in the file. On input, you can supply a session ID for a nonexistent session, with the result that the next record is read.

- NUM-WANTED is the number of sessions for which information is to be returned. In the response, this field is unchanged.

- RETN-CODE is the return code. TISERV returns a code in this field to indicate one of the following entries:
  
  0  OK
  4001  W-EOF
  4010  E-BAD-TRANSACTION
  4093  E-SECURITY-VIOLATION
  4904  E-ERR-SESSION-FILE

- RETN-CODE-DETAIL is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

- NUM-RETURNED is the number of sessions for which TRANSFER actually returns information.

- SESSION-DATA contains the correspondent name (CORR-NAME) and session ID (SESSION-ID) for the selected sessions. If the correspondent represented by your process has only read security privilege (and not write privilege), TISERV returns zero to the session ID field.

GET-NEXT-SESSION OPERATION. GET-NEXT-SESSION returns the names of correspondents with active sessions.

If the number of elements remaining in the list is less than or equal to the number requested, or if no active sessions were found, the warning W-EOF is returned in the RETN-CODE field.

A system administrator with write privilege at a node can terminate any session at that node by:

1. obtaining the session ID through the GET-NEXT-SESSION UOW
2. using this session ID in the IPC header and transmitting the END-SESSION UOW in the IPC message.
GET-PROFILE-ELEM (UOW Code 203)

GET-PROFILE-ELEM retrieves data elements from a depot Profile file. Typically, this UOW is issued before an ALTER-PROFILE-ELEM UOW.

DEFINITION I (a DDL skeleton format to which definitions can be added; for general applications use)

```plaintext
DEF get-profile-elem-usk.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED
                      VALUE 203.
    02 corr-name TYPE PIC X(80) VALUE SPACES.
END.

DEF get-profile-elem-rsk.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED
                      VALUE 203.
    02 retn-code TYPE BINARY 16.
    02 retn-code-detail TYPE BINARY 16.
    02 corr-name TYPE PIC X(80).
    02 num-returned TYPE BINARY 16.
END.
```

DEFINITION II (for TAL programs)

```plaintext
DEF get-profile-elem-uow.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED
                      VALUE 203.
    02 corr-name TYPE PIC X(80) VALUE SPACES.
    02 num-returned TYPE BINARY 16 UNSIGNED.
    02 elem-identifier TYPE BINARY 16
                      OCCURS 0 TO 10 TIMES
                      DEPENDING ON num-returned.
END.
```
GET-PROFILE-ELEM FIELDS. The fields defined in this UOW are:

- HDR is the UOW header. The UOW-CODE value is 203.

- CORR-NAME is the name of the correspondent whose depot profile will be retrieved. This is also the correspondent represented by the calling process. Only system administrators can enter correspondent names other than their own in this field; correspondents without system administrator privileges must enter their own names or leave the field blank. On input, you can use a partially qualified name, including wildcard characters. In the response, the fully qualified name is returned.

- NUM-RETURNED is the number of elements actually retrieved. This field contains:
  - the number retrieved - if no serious errors occurred
  - zero - if serious errors occurred

- ELEM-IDENTIFIER (used in Definition II only) specifies one or more element identifier numbers that indicate which data elements you wish to retrieve. You can specify any number, restricted only by the length of the entire IPC that contains this UOW. The maximum IPC length is defined during TRANSFER system configuration. The value 10 in the DDL noted for this field is not a limit.

- RETN-CODE is the return code. TISERV returns a code in this field to indicate one of the following entries.
To indicate successful retrieval:

0 OK

To indicate problems with the correspondent name:

5600 E-CORR-NSRV-ERR 5607 E-CORR-NSRV-DOWN
5601 E-CORR-NOT-FOUND 5611 E-CORR-NET-DOWN
5602 E-CORR-BAD-NAME 5622 E-CORR-NOT-SAME-NODE
5604 E-CORR-NO-SUCH-NODE 5623 E-CORR-AMBIGUOUS-NAME
5606 E-CORR-NSRV-NOT-FOUND

To indicate other problems:

4010 E-BAD TRANSACTION 4210 W-IDENTIFIER-ERRS
4093 E-SECURITY-VIOLATION 4902 E-ERR-PROFILE-FILE
4201 E-CONTEXT-ERR

- RETN-CODE-DETAIL is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

- ELEM-DATA-BLOCK (used in Definition II only) contains the specific fields retrieved, and is either 36 or 84 bytes long. The formats for blocks of these two lengths are:

  - For the short (36-byte) block

    DEF profile-elem-short.
    02 elem-identifier TYPE BINARY 16.
    02 elem-retn-code TYPE BINARY 16.
    02 elem-data TYPE CHARACTER 32.
    END.

  - For the long (84-byte) block

    DEF profile-elem-long.
    05 elem-identifier TYPE BINARY 16.
    05 elem-retn-code TYPE BINARY 16.
    05 elem-data TYPE CHARACTER 80.
    END.

In a single UOW, you can specify identifiers that return several element blocks. Long and short element blocks can be mixed.

Element blocks have the same format whether they appear in the GET-PROFILE-ELEM UOW or in the ALTER-PROFILE-ELEM UOW.
UOW Descriptions
GET-PROFILE-ELEM

GET-PROFILE-ELEM OPERATION. GET-PROFILE-ELEM retrieves one or more specific data elements from the Profile file. This UOW can be used by itself or in conjunction with the ALTER-PROFILE-ELEM UOW, which alters the data elements retrieved by GET-PROFILE-ELEM. Typically, you retrieve the data elements with the GET-PROFILE-ELEM UOW and then modify some or all of them with the ALTER-PROFILE-ELEM UOW. You can use the response from GET-PROFILE-ELEM as a request to ALTER-PROFILE-ELEM simply by changing the UOW code. These UOWs operate only on profile records controlled by TRANSFER, that is, those records with a REC-TYPE value less than 1000.

NOTE

Because these two UOWs are closely associated and because they share common fields, the operational details of both UOWs are discussed together in the following paragraphs. The structure of the ALTER-PROFILE-ELEM UOW, however, is presented earlier in this section.

Both the GET-PROFILE-ELEM and the ALTER-PROFILE-ELEM UOWs can be specified in either of two possible DDL definitions. Both of these definitions are discussed in the following paragraphs.

Definition I (for General Applications Use). The data structure provided by Definition I is convenient for most applications, and is simpler to use than Definition II. You typically construct a GET-PROFILE-ELEM UOW by specifying all fields up to and including NUM-RETURNED, and then specifying separate definitions for each particular element. Alternatively, you can define the element block as a DEPENDING ON construction using either PROFILE-ELEMENT-SHORT or PROFILE-ELEMENT-LONG; notice, however, that this type of construction does not permit mixing both long and short blocks.

When using either the GET-PROFILE-ELEM or ALTER-PROFILE-ELEM UOW, you specify those elements in the Profile file that you wish to retrieve or alter. For each element, the GET-PROFILE-ELEM UOW returns a block containing the unique numeric identifier of the element, a return code, and the data contained in the element.

In general, the GET-PROFILE-ELEM UOW and ALTER-PROFILE-ELEM UOW are used together in this way:

1. The GET-PROFILE-ELEM UOW returns data elements.
2. You can specify these same elements when calling the ALTER-PROFILE-ELEM UOW to alter the profile.
The coding involved is illustrated by the following examples:

Example 1

A GET-PROFILE-ELEM request and response is written to retrieve three elements: the depot privilege flags (ID 7), the current correspondent name (ID 1011), and the time the last session ended (ID 9). The ID numbers refer to the DDL data definitions for the Profile file data elements; these IDs and definitions are listed in a table that follows the GET-PROFILE-ELEM UOW discussion.

The COBOL coding would be:

```
01 myget-prof-elem-uow.
  copy get-profile-elem-usk of gcob.
    05 num-wanted PIC 9999 COMP VALUE 3.
    copy id-depot-privilege-flags of gcob.
    copy id-session-corr of gcob.
    copy id-depot-session-ended of gcob.

01 myget-prof-elem-rsp.
  copy get-profile-elem-rsk of gcob.
  copy data-depot-privilege-flags of gcob.
  copy data-session-corr of gcob.
  copy data-depot-session-ended of gcob.
```

The DDL equivalent would be:

```
DEF myget-profile-elem-uow.
  05 get-profile-elem-usk TYPE *.
  05 num-wanted TYPE BINARY 16 UNSIGNED VALUE 3.
  05 id-depot-privilege-flags TYPE *.
  05 id-session-corr TYPE *.
  05 id-depot-session-ended TYPE *.
END.

DEF myget-profile-elem-rsp.
  05 get-profile-elem-rsk TYPE *.
  05 data-depot-privilege-flags TYPE *.
  05 data-session-corr TYPE *.
  05 data-depot-session-ended TYPE *.
END.
```

Notice that a response from GET-PROFILE-ELEM has exactly the same structure as a request to ALTER-PROFILE-ELEM, as discussed in the next example. To change the response into a request, simply alter the UOW number in the UOW header.
Example 2

An ALTER-PROFILE-ELEM request and response is written for the same three elements specified in Example 1.

The COBOL coding would be:

```cobol
01 myalter-prof-elem-uow.
   copy alter-profile-elem-usek of gcob.
   copy data-depot-privilege-flags of gcob.
   copy data-session-corr of gcob.
   copy data-depot-session-ended of gcob.

01 myalter-prof-elem-rsp.
   copy alter-profile-elem-rsk of gcob.
   copy rtn-depot-privilege-flags of gcob.
   copy rtn-session-corr of gcob.
   copy rtn-depot-session-ended of gcob.
```

The DDL equivalent would be:

```ddl
DEF myalter-prof-elem-uow.
   05 alter-profile-elem-usk TYPE *.
   05 data-depot-privilege-flags TYPE *.
   05 data-session-corr. TYPE *.
   05 data-depot-session-ended. TYPE *.
END.

DEF myalter-prof-elem-rsp.
   05 alter-profile-elem-rsk TYPE *.
   05 rtn-depot-privilege-flags TYPE *.
   05 rtn-session-corr. TYPE *.
   05 rtn-depot-session-ended TYPE *.
END.
```

The storage layout for examples 1 and 2 is illustrated in Figure 5-6.
Figure 5-6. Storage Layout for GET-PROFILE-ELEM and ALTER-PROFILE-ELEM
Figure 5-6. Storage Layout for GET-PROFILE-ELEM and ALTER-PROFILE-ELEM (Continued)
Definition II (for TAL Programs). The data structure provided by Definition II is primarily intended for use in TAL programs.

When using either the GET-PROFILE-ELEM UOW or ALTER-PROFILE-ELEM UOW with Definition II, your application specifies a list of element identifiers. These codes identify the exact data elements wanted from the profile. For each identifier, the GET-PROFILE-ELEM UOW returns a block (defined in the ELEM-DATA-BLOCK field) containing the identifier, a return code, and the data requested. That data block is either 36 or 84 bytes long. Identifiers numbered less than 1000 use 36-byte blocks, while identifiers numbered 1000 or greater use 84-byte blocks.

Basically, the GET-PROFILE-ELEM UOW and ALTER-PROFILE-ELEM UOW are used together in this way:

1. The GET-PROFILE-ELEM UOW returns data elements in the ELEM-DATA-BLOCK field.

2. You can include this same ELEM-DATA-BLOCK field when calling the ALTER-PROFILE-ELEM UOW to alter the same data elements.

As previously noted, the format of ELEM-DATA-BLOCK, which really consists of individual definitions for each data element, is the same for both the GET-PROFILE-ELEM response UOW and the ALTER-PROFILE-ELEM UOW.

When the ALTER-PROFILE-ELEM UOW is invoked, TRANSFER first validates all modifications. If these modifications are valid, TRANSFER then performs them. The elem-data section of a modification group can be 32 or 80 bytes long, as implied by the particular identifier. The use of these two lengths simplifies COBOL record definition, as it allows the OCCURS DEPENDING ON construction. Note, however, that short and long data blocks cannot be mixed when you use the OCCURS DEPENDING ON construction. If the value requires fewer bytes than the reserved space, the value is placed at the left side of the field and the balance of the field is declared with a FILLER.

Depot Profile File Elements and Access Privileges. A list of the individual data elements in the depot Profile file appears in Table 5-1. This table gives the element identifier number as entered in ELEM-IDENTIFIER, the contents of the element, and the access privilege required for the GET-PROFILE-ELEM or ALTER-PROFILE-ELEM UOWs.

The following terms that appear in the table have the following definitions:

no one - The element cannot be accessed or updated.

anyone - The element can be accessed by anyone.
UOW Descriptions
GET-PROFILE-ELEM

owner - The element can be accessed only by the correspondent who owns the current session (in the case of a human correspondent, the logged-on user) or by a system administrator.

sys admin - For GET-PROFILE-ELEM, the element can be accessed only by a correspondent who has system administrator read-only privilege or write privilege. For ALTER-PROFILE-ELEM, the element can be accessed only by a correspondent who has system administrator write privilege.

automatic - The element is maintained automatically by TRANSFER. No one can alter the element, and the element identifier is invalid if used.
## Table 5-1. Depot Profile File Elements and Access Privileges

<table>
<thead>
<tr>
<th>Ident. No.</th>
<th>Content</th>
<th>GET-PROFILE-ELEM Privilege</th>
<th>ALTER-PROFILE-ELEM Privilege</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Depot password: old and new password</td>
<td>no one</td>
<td>owner, sys admin</td>
</tr>
<tr>
<td>2</td>
<td>Depot delta times: default delivery window for depot</td>
<td>owner, sys admin</td>
<td>owner, sys admin</td>
</tr>
<tr>
<td>3</td>
<td>Default package priority for depot</td>
<td>owner, sys admin</td>
<td>owner, sys admin</td>
</tr>
<tr>
<td></td>
<td>Maximum priority for depot</td>
<td>owner, sys admin</td>
<td>sys admin</td>
</tr>
<tr>
<td>4</td>
<td>Depot service flags: logon restriction, concurrent sessions, remote and local name resolution deferral; acceptance of suffixes</td>
<td>owner, sys admin</td>
<td>owner, sys admin (logon restricted flag modifiable by sys admin only, but sys admin cannot set the logon flag for own depot; thus, sys admin cannot disable self, leaving system without a sys admin)</td>
</tr>
<tr>
<td>5</td>
<td>Depot log flags: diagnostic logging</td>
<td>owner, sys admin</td>
<td>owner, sys admin</td>
</tr>
</tbody>
</table>
Table 5-1. Depot Profile File Elements and Access Privileges (Continued)

<table>
<thead>
<tr>
<th>Ident. No.</th>
<th>Content</th>
<th>GET-PROFILE-ELEM Privilege</th>
<th>ALTER-PROFILE-ELEM Privilege</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Depot privilege flags: system administrator capabilities</td>
<td>owner, sys admin</td>
<td>sys admin (but only for a depot other than own, so that sys admin cannot disable self and leave system without a sys admin)</td>
</tr>
<tr>
<td>8</td>
<td>Depot GUARDIAN ID: user name and password</td>
<td>owner, sys admin</td>
<td>owner, sys admin (plus matching password or group manager capability)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If GUARDIAN ID and password same as that under which TISERV is running, any password is accepted; password is verified when access to given GUARDIAN ID actually required</td>
</tr>
<tr>
<td>9</td>
<td>Last session ended: last session end/last session start times</td>
<td>owner, sys admin</td>
<td>automatic</td>
</tr>
<tr>
<td>10</td>
<td>Time that depot was created</td>
<td>owner, sys admin</td>
<td>automatic</td>
</tr>
</tbody>
</table>
Table 5-1. Depot Profile File Elements and Access Privileges (Continued)

<table>
<thead>
<tr>
<th>Ident. No.</th>
<th>Content</th>
<th>GET-PRIFILE-ELEM Privilege</th>
<th>ALTER-PRIFILE-ELEM Privilege</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>Time that depot was last updated by an ALTER-PRIFILE-ELEM UOW</td>
<td>owner</td>
<td>automatic</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sys admin</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Version code for depot Profile file</td>
<td>sys admin</td>
<td>automatic</td>
</tr>
<tr>
<td>13</td>
<td>System time window limits: maximum package lifespan, minimum delivery duration, and minimum time until expiration</td>
<td>anyone</td>
<td>sys admin</td>
</tr>
<tr>
<td>16</td>
<td>T/MAIL depot print-control flags: new page indication, and page indent</td>
<td>owner, sys admin</td>
<td>owner, sys admin</td>
</tr>
<tr>
<td>17</td>
<td>T/MAIL client operation flags: read-sequence determination, reply indicator, date of presentation criteria, and Virtual Screen use indicator</td>
<td>owner, sys admin</td>
<td>owner, sys admin</td>
</tr>
<tr>
<td>19</td>
<td>Default volume and subvolume for T/MAIL client</td>
<td>owner, sys admin</td>
<td>owner, sys admin</td>
</tr>
<tr>
<td>20</td>
<td>T/MAIL client system priority values</td>
<td>owner, anyone</td>
<td>sys admin</td>
</tr>
<tr>
<td>Ident. No.</td>
<td>Content</td>
<td>GET-PROFILE-ELEM Privilege</td>
<td>ALTER-PROFILE-ELEM Privilege</td>
</tr>
<tr>
<td>-----------</td>
<td>---------</td>
<td>---------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>26</td>
<td>Current administrator privileges: system administrator read and write privileges, and user-owner relation</td>
<td>logged-on correspondent</td>
<td>automatic</td>
</tr>
<tr>
<td>27</td>
<td>Dummy definition used for short ELEMENT-BLOCK</td>
<td>anyone</td>
<td>anyone</td>
</tr>
<tr>
<td>1003</td>
<td>T/MAIL client output file name</td>
<td>owner, sys admin</td>
<td>owner, sys admin</td>
</tr>
<tr>
<td>1005</td>
<td>FAX default destination</td>
<td>owner, sys admin</td>
<td>owner, sys admin</td>
</tr>
<tr>
<td>1006</td>
<td>FAX depot default folder</td>
<td>owner, sys admin</td>
<td>owner, sys admin</td>
</tr>
<tr>
<td>1011</td>
<td>Name of correspondent associated with current session</td>
<td>logged-on correspondent</td>
<td>automatic</td>
</tr>
<tr>
<td>1012</td>
<td>Dummy definition used for long ELEMENT-BLOCK</td>
<td>anyone</td>
<td>anyone</td>
</tr>
<tr>
<td>1013</td>
<td>Name of correspondent who last updated profile</td>
<td>owner, sys admin</td>
<td>automatic</td>
</tr>
</tbody>
</table>
Data Element DDL and Return Codes. The DDL definitions for each specific data element, as contained in ELEM-DATA-BLOCK, and the return codes supplied by the GET-PROFILE-ELEM UOW appear in Table 5-2.

The ALTER-PROFILE-ELEM UOW does not supply return codes within the ELEM-DATA-BLOCK field. Instead, this UOW provides its return codes in the ELEM-RETN-CODE field of its response UOW. These return codes are listed in Table 5-3.
<table>
<thead>
<tr>
<th>Ident.</th>
<th>No.</th>
<th>Definition and Field Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 DEF</td>
<td>data-depot-password.</td>
<td>TYPE BINARY 16 VALUE 1.</td>
</tr>
<tr>
<td>02 elem-identifier</td>
<td>TYPE BINARY 16.</td>
<td></td>
</tr>
<tr>
<td>02 dd-password-rtn</td>
<td>TYPE CHARACTER 16.</td>
<td></td>
</tr>
<tr>
<td>02 old-password</td>
<td>TYPE CHARACTER 16.</td>
<td></td>
</tr>
<tr>
<td>02 new-password</td>
<td>END.</td>
<td></td>
</tr>
</tbody>
</table>

**FIELDS**

- **ELEM-IDENTIFIER** is 1.
- **DD-PASSWORD-RTN** is E-INVALID-IDENTIFIER.
- **OLD-PASSWORD** is the password prior to the invoking of ALTER-PROFILE-ELEM. System administrators can modify their own passwords and those of other users.
- **NEW-PASSWORD** is the new password assigned by ALTER-PROFILE-ELEM. If both the old password and the new password are specified as blanks, the depot password is not changed.
  - Depot owners can change their passwords by setting the OLD-PASSWORD field to the correct password of the depot and setting the NEW-PASSWORD field to the desired new password.
  - System administrators can change the password of any depot other than their own by setting the OLD-PASSWORD field to blanks and the NEW-PASSWORD field to the desired new password.
### Table 5-2. DDL Data Definitions for Profile File Data Elements with GET-PROFILE-ELEM Return Codes (Continued)

<table>
<thead>
<tr>
<th>Ident. No.</th>
<th>Definition and Field Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 DEF data-depot-deltas.</td>
<td>TYPE BINARY 16 VALUE 2.</td>
</tr>
<tr>
<td>02 elem-identifier</td>
<td>TYPE BINARY 16.</td>
</tr>
<tr>
<td>02 dd-deltas-rtn</td>
<td>TYPE CHARACTER 20 VALUE SPACES.</td>
</tr>
<tr>
<td>02 deliv-start-delta.</td>
<td>TYPE CHARACTER 20 VALUE SPACES.</td>
</tr>
<tr>
<td>03 quantity</td>
<td>PIC 9(4) COMP.</td>
</tr>
<tr>
<td>03 units</td>
<td>PIC A.</td>
</tr>
<tr>
<td>03 filler</td>
<td>PIC X.</td>
</tr>
<tr>
<td>02 deliv-end-delta.</td>
<td>PIC 9(4) COMP.</td>
</tr>
<tr>
<td>03 quantity</td>
<td>PIC A.</td>
</tr>
<tr>
<td>03 units</td>
<td>PIC X.</td>
</tr>
<tr>
<td>02 expiration-delta.</td>
<td>PIC 9(4) COMP.</td>
</tr>
<tr>
<td>03 quantity</td>
<td>PIC A.</td>
</tr>
<tr>
<td>03 units</td>
<td>PIC X.</td>
</tr>
<tr>
<td>02 filler</td>
<td>TYPE CHARACTER 20 VALUE SPACES.</td>
</tr>
</tbody>
</table>

### FIELDS
- **ELEM-IDENTIFIER** is 2.
- **DELTAS-RTN** is one of the following:
  
  0 OK
  4238 E-TFER-PROFILE-ABSENT

- **DELIV-START-DELTA** is the default period between package submittal and the earliest time that delivery can begin.
- **QUANTITY** is a numeric increment of the time unit specified by UNITS, which can be D (for days), H (for hours), or M (for minutes).
Table 5-2. DDL Data Definitions for Profile File Data Elements with GET-PROFILE-ELEM Return Codes (Continued)

<table>
<thead>
<tr>
<th>Ident. No.</th>
<th>Definition and Field Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>The UOW always returns the largest time unit, while still retaining whole quantities of those units. For example:</td>
</tr>
<tr>
<td>cont'd</td>
<td>An update of 60 M returns 1 H.</td>
</tr>
<tr>
<td></td>
<td>An update of 75 M returns 75 M.</td>
</tr>
<tr>
<td></td>
<td>• DELIV-END-DELTA is the default time period between initiation of delivery and latest permitted delivery time.</td>
</tr>
<tr>
<td></td>
<td>• EXPIRATION-DELTA is the default time period between delivery end and the time when the package expires.</td>
</tr>
</tbody>
</table>

3 DEF data-depot-priorities.
   02 elem-identifier TYPE BINARY 16 VALUE 3.
   02 dd-priorities-rtn TYPE BINARY 16.
   02 default-priority PIC 9(3) COMP.
   02 maximum-priority PIC 9(3) COMP.
   02 filler TYPE CHARACTER 28 VALUE SPACES.
END.

FIELDS
- ELEMENT-IDENTIFIER is 3.
- DD-PRIORITIES-RTN is one of the following:
  
  0 OK
  4238 E-TFER-PROFILE-ABSENT

- DEFAULT-PRIORITY is the default priority assigned to packages submitted from the depot. (T/MAIL has its own default priority; see Identifier 20.)
Table 5-2. DDL Data Definitions for Profile File Data Elements with GET-PROFILE-ELEM Return Codes (Continued)

<table>
<thead>
<tr>
<th>Ident. No.</th>
<th>Definition and Field Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>• MAXIMUM-PRIORITY is the maximum priority that can be assigned to packages submitted from the depot. Only a system administrator can change this field.</td>
</tr>
</tbody>
</table>

4 DEF data-depot-service-flags.
   02 element-identifier TYPE BINARY 16 VALUE 4.
   02 dd-service-flags-retn TYPE BINARY 16.
   02 logon-restricted-flag TYPE BOOLEAN.
   02 reserved-flag2 TYPE BOOLEAN VALUE "N".
   02 concurrent-session-flag TYPE BOOLEAN.
   02 defer-remote-flag TYPE BOOLEAN.
   02 defer-local-flag TYPE BOOLEAN.
   02 suffixes-accepted-flag TYPE BOOLEAN.
   02 reserved-flag7 TYPE BOOLEAN VALUE "N".
   02 reserved-flag8 TYPE BOOLEAN VALUE "N".
   02 filler TYPE CHARACTER 24 VALUE SPACES.
END.

FIELDS
• ELEMENT-IDENTIFIER is 4.
• DD-SERVICE-FLAGS-RTN is one of the following:
  0 OK
  4238 E-TFER-PROFILE-ABSENT
• LOGON-RESTRICTED-FLAG determines whether anyone can logon at the depot.
  N = prohibit logon
  Y = permit logon
This field can only be altered by system administrators, and only for a depot other than their own.
• RESERVED-FLAG2 is reserved for use by Tandem; you must set this field to N.
Table 5-2. DDL Data Definitions for Profile File Data Elements with GET-PROFILE-ELEM Return Codes (Continued)

<table>
<thead>
<tr>
<th>Ident. No.</th>
<th>Definition and Field Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td><strong>CONCURRENT-SESSION-FLAG</strong> determines whether more than one session can be conducted simultaneously at the depot.</td>
</tr>
<tr>
<td></td>
<td><strong>Y</strong> = permit concurrent sessions</td>
</tr>
<tr>
<td></td>
<td><strong>N</strong> = prohibit concurrent sessions</td>
</tr>
<tr>
<td></td>
<td>When TRANSFER checks for concurrent sessions, sessions created for agents are not counted in this verification.</td>
</tr>
<tr>
<td></td>
<td><strong>DEFER-REMOTE-FLAG</strong> determines whether the resolution of remote names is deferred at the depot.</td>
</tr>
<tr>
<td></td>
<td><strong>Y</strong> = defer resolution</td>
</tr>
<tr>
<td></td>
<td><strong>N</strong> = permit immediate resolution</td>
</tr>
<tr>
<td></td>
<td>This flag is used during the processing of the ADD-RECIP and ADD-MEMBER UOWs.</td>
</tr>
<tr>
<td></td>
<td><strong>DEFER-LOCAL-FLAG</strong> determines whether the resolution of local names is deferred at the depot.</td>
</tr>
<tr>
<td></td>
<td><strong>Y</strong> = defer resolution</td>
</tr>
<tr>
<td></td>
<td><strong>N</strong> = permit immediate resolution</td>
</tr>
<tr>
<td></td>
<td>This flag is used in special cases where a client wishes to add a recipient name or member name even if that name does not exist.</td>
</tr>
</tbody>
</table>

**WARNING**

The **DEFER-LOCAL-FLAG** is normally set to **N**. You should change this setting only after consultation with your Tandem system analyst.
Table 5-2. DDL Data Definitions for Profile File Data Elements with GET-PROFILE-ELEM Return Codes (Continued)

<table>
<thead>
<tr>
<th>Ident. No.</th>
<th>Definition and Field Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>• SUFFIXES-ACCEPTED-FLAG determines whether the depot recognizes suffixes on incoming packages.</td>
</tr>
<tr>
<td></td>
<td>N = ignore suffixes. Any suffixes in recipient lists or distribution lists that reference this correspondent will be ignored; and any packages will be delivered to this depot only once and need to be acknowledged by this depot only once.</td>
</tr>
<tr>
<td></td>
<td>Y = allow suffixes. Packages will be delivered to this depot once for each different suffix used in recipient lists or distribution lists that reference this depot. For each delivery, TRANSFER will save a package in the depot INBOX, but only if the package is not already there, and invoke any appropriate agents. Similarly, the package must be acknowledged for each different suffix that was used in recipient lists or distribution lists.</td>
</tr>
<tr>
<td></td>
<td>• RESERVED-FLAG7 and RESERVED-FLAG8 are reserved for use by Tandem; you must set these fields to N.</td>
</tr>
</tbody>
</table>

```
5 DEF data-depot-log-flags.
  02 element-identifier TYPE BINARY 16 VALUE 5.
  02 dd-log-flags-rtn TYPE BINARY 16.
  02 diagnostic-log-flag TYPE BOOLEAN.
  02 reserved-flag2 TYPE BOOLEAN VALUE "N".
  02 reserved-flag3 TYPE BOOLEAN VALUE "N".
  02 reserved-flag4 TYPE BOOLEAN VALUE "N".
  02 reserved-flag5 TYPE BOOLEAN VALUE "N".
  02 reserved-flag6 TYPE BOOLEAN VALUE "N".
  02 reserved-flag7 TYPE BOOLEAN VALUE "N".
  02 reserved-flag8 TYPE BOOLEAN VALUE "N".
  02 filler TYPE CHARACTER 24 VALUE SPACES.
END.
```
Table 5-2. DDL Data Definitions for Profile File Data Elements with GET-PROFILE-ELEM Return Codes (Continued)

<table>
<thead>
<tr>
<th>No.</th>
<th>Definition and Field Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 FIELDS cont'd</td>
<td></td>
</tr>
<tr>
<td>• ELEM-IDENTIFIER is 5.</td>
<td></td>
</tr>
<tr>
<td>• DD-LOG-FLAGS-RTN is one of the following:</td>
<td></td>
</tr>
<tr>
<td>0 OK</td>
<td></td>
</tr>
<tr>
<td>4238 E-TFER-PROFILE-ABSENT</td>
<td></td>
</tr>
<tr>
<td>• DIAGNOSTIC-LOG-FLAG determines whether diagnostic logging of IPCs takes place.</td>
<td></td>
</tr>
<tr>
<td>Y = enables logging even if systemwide logging is not enabled</td>
<td></td>
</tr>
<tr>
<td>N = disables logging, unless systemwide logging is enabled</td>
<td></td>
</tr>
<tr>
<td>• RESERVED-FLAG2 through RESERVED-FLAG8 are reserved for use by Tandem. You must set these fields to N.</td>
<td></td>
</tr>
</tbody>
</table>

| 7 DEF data-depot-privilege-flags. |
| 02 elem-identifier TYPE BINARY 16 VALUE 7. |
| 02 dd-privilege-flags-rtn TYPE BINARY 16. |
| 02 sys-admin-flag TYPE CHARACTER 1. |
| 02 filler TYPE CHARACTER 31 VALUE SPACES. |
| END. |

FIELDS |
• ELEM-IDENTIFIER is 7. |
• DD-PRIVILEGE-FLAGS-RTN is one of the following: |
|   0 OK |
|   4238 E-TFER-PROFILE-ABSENT |
Table 5-2. DDL Data Definitions for Profile File Data Elements
with GET-PROFILE-ELEM Return Codes (Continued)

<table>
<thead>
<tr>
<th>Ident. No.</th>
<th>Definition and Field Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 • SYS-ADMIN-FLAG determines what, if any, system administrator privileges the correspondent has.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[ W = write privileges ]</td>
</tr>
<tr>
<td></td>
<td>[ R = read privileges ]</td>
</tr>
<tr>
<td></td>
<td>[ N = the correspondent is not a system administrator ]</td>
</tr>
<tr>
<td></td>
<td>If system administrators attempt to modify their own flags, a security violation results.</td>
</tr>
</tbody>
</table>

| 8 DEF data-depot-guardian-id. |
| 02 elem-identifier TYPE BINARY 16 VALUE 8. |
| 02 dd-guardian-id-rtn TYPE BINARY 16. |
| 02 guardian-user-name TYPE CHARACTER 17. |
| 02 guardian-password TYPE CHARACTER 8. |
| 02 filler TYPE CHARACTER 7 VALUE SPACES. |
END. |

FIELDS |
• ELEM-IDENTIFIER is 8. |
• DD-GUARDIAN-ID-RTN is one of the following: |
  0 OK |
  4238 E-TFER-PROFILE-ABSENT |
• GUARDIAN-USER-NAME is the correspondent user name as defined to the GUARDIAN operating system. If no GUARDIAN user name is associated with the depot, the entry NONE is returned in this field.
Table 5-2. DDL Data Definitions for Profile File Data Elements with GET-PROFILE-ELEM Return Codes (Continued)

<table>
<thead>
<tr>
<th>Ident. No.</th>
<th>Definition and Field Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>GUARDIAN-PASSWORD is the correspondent logon password required for validation of an altered GUARDIAN ID. If the GUARDIAN user name is being changed, the GUARDIAN password must be supplied. Under the following conditions, the password is accepted without verification:</td>
</tr>
<tr>
<td>cont'd</td>
<td>- if the depot requesting the change has a GUARDIAN user ID of group.MANAGER, where group is the group of the new GUARDIAN ID being assigned to the depot</td>
</tr>
<tr>
<td></td>
<td>- if the depot requesting the change has a GUARDIAN user ID of SUPER.SUPER.</td>
</tr>
<tr>
<td></td>
<td>This simulates the validation performed by the GUARDIAN operating system. GUARDIAN-PASSWORD is used only by the ALTER-PROFILE-ELEM UOW; blanks are returned in this field for GET-PROFILE-ELEM.</td>
</tr>
<tr>
<td></td>
<td>If GUARDIAN-USER-NAME is not being changed, the password is ignored. To change a depot GUARDIAN password, first change the GUARDIAN-USER-NAME to NONE with blanks in the password; then change GUARDIAN-USER-NAME to the original user name with the new password.</td>
</tr>
</tbody>
</table>
### Table 5-2. DDL Data Definitions for Profile File Data Elements with GET-PROFILE-ELEM Return Codes (Continued)

<table>
<thead>
<tr>
<th>Ident. No.</th>
<th>Definition and Field Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>9 DEF</td>
<td>data-depot-session-ended.</td>
</tr>
<tr>
<td></td>
<td>02 elem-identifier</td>
</tr>
<tr>
<td></td>
<td>02 dd-session-ended-rtn</td>
</tr>
<tr>
<td></td>
<td>02 session-ended.</td>
</tr>
<tr>
<td></td>
<td>03 year</td>
</tr>
<tr>
<td></td>
<td>03 month</td>
</tr>
<tr>
<td></td>
<td>03 day-of-month</td>
</tr>
<tr>
<td></td>
<td>03 hour</td>
</tr>
<tr>
<td></td>
<td>03 minute</td>
</tr>
<tr>
<td></td>
<td>03 second</td>
</tr>
<tr>
<td></td>
<td>02 session-started.</td>
</tr>
<tr>
<td></td>
<td>03 year</td>
</tr>
<tr>
<td></td>
<td>03 month</td>
</tr>
<tr>
<td></td>
<td>03 day-of-month</td>
</tr>
<tr>
<td></td>
<td>03 hour</td>
</tr>
<tr>
<td></td>
<td>03 minute</td>
</tr>
<tr>
<td></td>
<td>03 second</td>
</tr>
<tr>
<td></td>
<td>02 filler</td>
</tr>
</tbody>
</table>

END.

**FIELDS**

- **ELEM-IDENTIFIER** is 9.
- **DD-SESSION-ENDED-RETN** is one of the following:
  
  0 OK
  4238 E-TFER-PROFILE-ABSENT

- **SESSION-ENDED** contains the date and time that the last session for the depot ended. This excludes sessions automatically initiated by TRANSFER for an agent.
- **SESSION-STARTED** contains the date and time that the last session began.
Table 5-2. DDL Data Definitions for Profile File Data Elements with GET-PROFILE-ELEM Return Codes (Continued)

<table>
<thead>
<tr>
<th>No.</th>
<th>Definition and Field Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>DEF data-depot-created-time.</td>
</tr>
<tr>
<td></td>
<td>02 elem-identifier</td>
</tr>
<tr>
<td></td>
<td>02 dd-created-time-rtn</td>
</tr>
<tr>
<td></td>
<td>02 depot-created.</td>
</tr>
<tr>
<td></td>
<td>03 year</td>
</tr>
<tr>
<td></td>
<td>03 month</td>
</tr>
<tr>
<td></td>
<td>03 day-of-month</td>
</tr>
<tr>
<td></td>
<td>03 hour</td>
</tr>
<tr>
<td></td>
<td>03 minute</td>
</tr>
<tr>
<td></td>
<td>03 second</td>
</tr>
<tr>
<td></td>
<td>02 version-created.</td>
</tr>
<tr>
<td></td>
<td>03 letter</td>
</tr>
<tr>
<td></td>
<td>03 rev-number</td>
</tr>
<tr>
<td></td>
<td>02 filler</td>
</tr>
<tr>
<td></td>
<td>END.</td>
</tr>
</tbody>
</table>

FIELDS

- ELEM-IDENTIFIER is 10.

- DD-CREATED-TIME-RTN is one of the following:
  
  0 OK
  4238 E-TFER-PROFILE-ABSENT

- DEPOT-CREATED is the date and time of the depot creation.

- VERSION-CREATED is the version of TRANSFER that created the depot. For the first release of the product, LETTER is A and REV-NUMBER is 01.
Table 5-2. DDL Data Definitions for Profile File Data Elements with GET-PROFILE-ELEM Return Codes (Continued)

<table>
<thead>
<tr>
<th>Ident. No.</th>
<th>Definition and Field Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>DEF data-depot-updated-time.</td>
</tr>
<tr>
<td>02</td>
<td>elem-identifier TYPE BINARY 16 VALUE 11.</td>
</tr>
<tr>
<td>02</td>
<td>dd-updated-time-rtn TYPE BINARY 16.</td>
</tr>
<tr>
<td>02</td>
<td>depot-updated.</td>
</tr>
<tr>
<td>03</td>
<td>year PIC 9(4).</td>
</tr>
<tr>
<td>03</td>
<td>month PIC 9(2).</td>
</tr>
<tr>
<td>03</td>
<td>day-of-month PIC 9(2).</td>
</tr>
<tr>
<td>03</td>
<td>hour PIC 9(2).</td>
</tr>
<tr>
<td>03</td>
<td>minute PIC 9(2).</td>
</tr>
<tr>
<td>03</td>
<td>second PIC 9(2).</td>
</tr>
<tr>
<td>02</td>
<td>filler TYPE CHARACTER 18 VALUE SPACES.</td>
</tr>
<tr>
<td></td>
<td>END.</td>
</tr>
</tbody>
</table>

FIELDS

- ELEM-IDENTIFIER is 11.
- DD-UPDATED-TIME-RTN is one of the following:
  - 0 OK
  - 4238 E-TFER-PROFILE-ABSENT
- DEPOT-UPDATED is the date and time that the depot was last updated with the ALTER-PROFILE-ELEM UOW.

| 12         | DEF data-sys-file-version.       |
| 02         | elem-identifier TYPE BINARY 16 VALUE 12. |
| 02         | ds-file-version-rtn TYPE BINARY 16. |
| 02         | version-code.                    |
| 03         | letter PIC A.                    |
| 03         | rev-number PIC 99.               |
| 02         | filler TYPE CHARACTER 29 VALUE SPACES. |
|            | END.                             |

FIELDS

- ELEM-IDENTIFIER is 12.
Table 5-2. DDL Data Definitions for Profile File Data Elements with GET-PROFILE-ELEM Return Codes (Continued)

<table>
<thead>
<tr>
<th>Ident.</th>
<th>No.</th>
<th>Definition and Field Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>DS-FILE-VERSION-RTN is one of the following:</td>
<td>cont'd</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>OK</td>
</tr>
<tr>
<td></td>
<td>4241</td>
<td>E-TFER-CTL-REC-ABSENT</td>
</tr>
<tr>
<td></td>
<td>• VERSION-CODE is the current version code for the Profile file.</td>
<td>For the first release of the product, LETTER is A and REV-NUMBER is 01.</td>
</tr>
</tbody>
</table>

13 DEF data-sys-window-limits.
   02 elem-identifier TYPE BINARY 16 VALUE 13.
   02 ds-window-limits-rtn TYPE BINARY 16.
   02 max-lifespan.
      03 quantity PIC 9(4) COMP.
      03 units PIC A.
      03 filler PIC X.
   02 min-deliv.
      03 quantity PIC 9(4) COMP.
      03 units PIC A.
      03 filler PIC X.
   02 min-expire.
      03 quantity PIC 9(4) COMP.
      03 units PIC A.
      03 filler PIC X.
   02 filler TYPE CHARACTER 20 VALUE SPACES.

END.

FIELDS
- ELEM-IDENTIFIER is 13.
- DS-WINDOW-LIMITS-RTN is one of the following:
  0   OK
  4241 E-TFER-CTL-REC-ABSENT
<table>
<thead>
<tr>
<th>Ident. No.</th>
<th>Definition and Field Descriptions</th>
</tr>
</thead>
</table>
| 13 | **MAX-LIFESPAN** is the maximum lifespan of packages submitted from any depot at the node.  
   MIN-DELIV is the minimum delivery window (END-DELIV-TIME minus START-DELIV-TIME) that is allowed for a package.  
   MIN-EXPIRE is the minimum period allowed between the latest delivery time and the expiration time for a package.  
   A submitted package that violates these last two limits is adjusted to conform to these limits and is not rejected.  
   Updates to these fields will not affect any sessions currently in progress. |
| 16 | **ELEM-IDENTIFIER** is 16.  
   **DDM-PRINT-CTL-RTN** is one of the following:  
   0 OK  
   4239 E-MAIL-PROFILE-ABSENT |

Table 5-2. DDL Data Definitions for Profile File Data Elements with GET-PROFILE-ELEM Return Codes (Continued)
<table>
<thead>
<tr>
<th>Ident. No.</th>
<th>Definition and Field Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>PAGE-FLAG indicates whether each package is displayed by the T/MAIL client on a new page when the Print function key is pressed.</td>
</tr>
<tr>
<td></td>
<td>Y = new page</td>
</tr>
<tr>
<td></td>
<td>N = no new page</td>
</tr>
<tr>
<td></td>
<td>PAGE-INDENT-COLUMNS indicates the number of columns to the right that T/MAIL indents the page.</td>
</tr>
</tbody>
</table>

17 DEF data-depot-mail-flags.
   02 elem-identifier       TYPE BINARY 16 VALUE 17.
   02 ddm-flags-rtn         TYPE BINARY 16.
   02 read-sequence-flag    TYPE CHARACTER 1.
   02 reply-default-flag    TYPE CHARACTER 1.
   02 date-presentation-flag TYPE CHARACTER 1.
   02 edit-flag             TYPE BOOLEAN.
   02 filler                TYPE CHARACTER 28 VALUE SPACES.
END.

FIELDS

- ELEM-IDENTIFIER is 17.
- DDM-FLAGS-RTN is one of the following:
  0  OK
  4239  E-MAIL-PROFILE-ABSENT

- READ-SEQUENCE-FLAG indicates how packages sent to T/MAIL are to be read upon receipt.
  S = read in the order transmitted by the sender
  L = read the latest package delivered first
  T = read the earliest package delivered first
Table 5-2. DDL Data Definitions for Profile File Data Elements with GET-PROFILE-ELEM Return Codes (Continued)

<table>
<thead>
<tr>
<th>Ident. No.</th>
<th>Definition and Field Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>17 cont'd</td>
<td>NOTE</td>
</tr>
<tr>
<td></td>
<td>For TRANSFER B00, the READ-SEQUENCE-FLAG field has become obsolete for the TRANSFER ADMIN screen interface and T/MAIL.</td>
</tr>
<tr>
<td></td>
<td>For downward compatibility, the TISERV UOW interface will continue to support this field.</td>
</tr>
<tr>
<td></td>
<td>• REPLY-DEFAULT-FLAG indicates whom T/MAIL should place in the recipient list in response to the Reply function key.</td>
</tr>
<tr>
<td></td>
<td>T = the original package's To: recipients</td>
</tr>
<tr>
<td></td>
<td>C = all recipients copied</td>
</tr>
<tr>
<td></td>
<td>N = no one except the original package sender</td>
</tr>
<tr>
<td></td>
<td>• DATE-PRESENTATION indicates the format in which the date presented on packages sent from the depot should be displayed.</td>
</tr>
<tr>
<td></td>
<td>D = world format (dd/mm/yy)</td>
</tr>
<tr>
<td></td>
<td>M = USA format (mm/dd/yy)</td>
</tr>
<tr>
<td></td>
<td>• EDIT-FLAG indicates whether Edit VS (Virtual Screen) is automatically invoked if EDIT is being used.</td>
</tr>
<tr>
<td></td>
<td>Y = automatically invoke Edit VS</td>
</tr>
<tr>
<td></td>
<td>N = do not automatically invoke Edit VS</td>
</tr>
</tbody>
</table>

<p>| 19 DEF data-depot-mail-volumes. |
| 02 elem-identifier TYPE BINARY 16 VALUE 19. |
| 02 ddm-volumes-rtn TYPE BINARY 16. |
| 02 node TYPE CHARACTER 8. |
| 02 volume-subvolume TYPE CHARACTER 17. |
| 02 filler TYPE CHARACTER 7. VALUE SPACES. |
| END. |</p>
<table>
<thead>
<tr>
<th>Ident. No.</th>
<th>Definition and Field Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>19</strong> FIELDS cont'd</td>
<td></td>
</tr>
<tr>
<td>• ELEM-IDENTIFIER is 19.</td>
<td></td>
</tr>
<tr>
<td>• DDM-VOLUME-RTN is one of the following:</td>
<td></td>
</tr>
<tr>
<td>0 OK</td>
<td></td>
</tr>
<tr>
<td>4239 E-MAIL-PROFILE-ABSENT</td>
<td></td>
</tr>
<tr>
<td>• NODE is the name of the node at which the T/MAIL client's default volume is defined. This entry is written in the GUARDIAN/EXPAND format, where the node name begins with a backslash ().</td>
<td></td>
</tr>
<tr>
<td>• VOLUME-SUBVOLUME is the volume and subvolume that is to be the T/MAIL client's default volume, subvolume. If either the volume or subvolume, or both, are omitted, the servers use the defaults for the GUARDIAN ID. If the system name is omitted, the servers use the GUARDIAN default system.</td>
<td></td>
</tr>
<tr>
<td><strong>20</strong> DEF data-sys-mail-priority.</td>
<td></td>
</tr>
<tr>
<td>02 elem-identifier TYPE BINARY 16 VALUE 20.</td>
<td></td>
</tr>
<tr>
<td>02 dsm-priority-rtn TYPE BINARY 16.</td>
<td></td>
</tr>
<tr>
<td>02 low-priority PIC 9(3) COMP.</td>
<td></td>
</tr>
<tr>
<td>02 normal-priority PIC 9(3) COMP.</td>
<td></td>
</tr>
<tr>
<td>02 urgent-priority PIC 9(3) COMP.</td>
<td></td>
</tr>
<tr>
<td>02 filler TYPE CHARACTER 26 VALUE SPACES.</td>
<td></td>
</tr>
<tr>
<td>END.</td>
<td></td>
</tr>
<tr>
<td>FIELDS</td>
<td></td>
</tr>
<tr>
<td>• ELEM-IDENTIFIER is 20.</td>
<td></td>
</tr>
<tr>
<td>• DSM-PRIORITY-RTN is one of the following:</td>
<td></td>
</tr>
<tr>
<td>0 OK</td>
<td></td>
</tr>
<tr>
<td>4239 E-MAIL-PROFILE-ABSENT</td>
<td></td>
</tr>
</tbody>
</table>
Table 5-2. DDL Data Definitions for Profile File Data Elements with GET-PROFILE-ELEM Return Codes (Continued)

<table>
<thead>
<tr>
<th>Ident. No.</th>
<th>Definition and Field Descriptions</th>
</tr>
</thead>
</table>
| 20 cont'd  | • LOW-PRIORITY is a numeric value that represents low priority for the T/Mail client.  
|            | • NORMAL-PRIORITY is a numeric value that represents normal priority for the T/MAIL client.  
|            | • URGENT-PRIORITY is a numeric value that represents urgent priority for the T/MAIL client. This value cannot exceed the maximum priority for the depot.  
|            | LOW-PRIORITY, NORMAL-PRIORITY, and URGENT-PRIORITY can have the same values. |
| 26         | DEF data-session-privileges.  
| 02         | elem-identifier TYPE BINARY 16 VALUE 26.  
| 02         | ds-privileges-rtn TYPE BINARY 16.  
| 02         | syst-admin-flag TYPE CHARACTER 1.  
| 02         | user-owner-flag TYPE BOOLEAN.  
| 02         | filler TYPE CHARACTER 30 VALUE SPACES.  
| END        | FIELDS  
|            | • ELEMENT-IDENTIFIER is 26.  
|            | • DS-PRIVILEGES-RTN is OK.  
|            | • SYSTEM-ADMIN-FLAG indicates the correspondent's system administrator privileges.  
|            | W = write and read privileges  
|            | R = read privileges  
|            | N = no system administrator privileges |
Table 5-2. DDL Data Definitions for Profile File Data Elements with GET-PROFILE-ELEM Return Codes (Continued)

<table>
<thead>
<tr>
<th>Ident. No.</th>
<th>Definition and Field Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>26</td>
<td>• USER-OWNER-FLAG indicates whether the current user is the owner of the depot.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```
27 DEF data-dummy-short.
  02 elem-identifier TYPE BINARY 16 VALUE 27.
  02 dd-short-rtn TYPE BINARY 16.
  02 filler TYPE CHARACTER 32 VALUE 27.
END.

FIELDS
• ELEM-IDENTIFIER is 27.
• DD-SHORT-RTN is OK.
```

```
1003 DEF data-depot-mail-filename.
  02 elem-identifier TYPE BINARY 16 VALUE 1003.
  02 ddm-filename-retn TYPE BINARY 16.
  02 mail-output-filename PIC X(80).
END.

FIELDS
• ELEM-IDENTIFIER is 1003.
• DDM-FILENAME-RETN is one of the following:
  0 OK
  4239 E-MAIL-PROFILE-ABSENT
• MAIL-OUTPUT-FILENAME is the GUARDIAN file name of the output file used by the T/MAIL client.
```
Table 5-2. DDL Data Definitions for Profile File Data Elements with GET-PROFILE-ELEM Return Codes (Continued)

<table>
<thead>
<tr>
<th>Ident. No.</th>
<th>Definition and Field Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1005</td>
<td>DEF data-depot-fax-routing.</td>
</tr>
<tr>
<td></td>
<td>02  elem-identifier  TYPE BINARY 16</td>
</tr>
<tr>
<td></td>
<td>VALUE 1005.</td>
</tr>
<tr>
<td></td>
<td>02  ddf-routing-rtn  TYPE BINARY 16.</td>
</tr>
<tr>
<td></td>
<td>02  default-fax-routing  TYPE CHARACTER 80.</td>
</tr>
</tbody>
</table>

**FIELDS**

- ELEM-IDENTIFIER is 1005.
- DDF-ROUTING-RTN is one of the following:
  
  0  OK
  4240  E-FAX-PROFILE-ABSENT

- DEFAULT-FAX-ROUTING is used to maintain a default destination (a correspondent or distribution list name), for facsimile items without a destination address.

<table>
<thead>
<tr>
<th>1006</th>
<th>DEF data-depot-fax-folder1.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>02  elem-identifier  TYPE BINARY 16</td>
</tr>
<tr>
<td></td>
<td>VALUE 1006.</td>
</tr>
<tr>
<td></td>
<td>02  ddf-folder-rtn  TYPE BINARY 16.</td>
</tr>
<tr>
<td></td>
<td>02  default-fax-folder1  TYPE CHARACTER 80.</td>
</tr>
</tbody>
</table>

**FIELDS**

- ELEM-IDENTIFIER is 1006.
- DDF-FOLDER-RTN is one of the following:
  
  0  OK
  4240  E-FAX-PROFILE-ABSENT

- DEFAULT-FAX-FOLDER1 is used to maintain the name of the default folder into which facsimile items without destination addresses can be saved.
### Table 5-2. DDL Data Definitions for Profile File Data Elements with GET-PROFILE-ELEM Return Codes (Continued)

<table>
<thead>
<tr>
<th>Ident. No.</th>
<th>Definition and Field Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1011</td>
<td>DEF data-session-corr.</td>
</tr>
<tr>
<td></td>
<td>02 elem-identifier</td>
</tr>
<tr>
<td></td>
<td>02 ds-corr-rtn</td>
</tr>
<tr>
<td></td>
<td>02 session-corr-name</td>
</tr>
<tr>
<td></td>
<td>TYPE BINARY 16</td>
</tr>
<tr>
<td></td>
<td>VALUE 1011.</td>
</tr>
<tr>
<td></td>
<td>TYPE BINARY 16</td>
</tr>
<tr>
<td></td>
<td>PIC X(80).</td>
</tr>
<tr>
<td></td>
<td>FIELDS</td>
</tr>
<tr>
<td></td>
<td>• ELEM-IDENTIFIER is 1011.</td>
</tr>
<tr>
<td></td>
<td>• DS-CORR-RTN is OK.</td>
</tr>
<tr>
<td></td>
<td>• SESSION-CORR-NAME is the fully qualified name of the logged-on correspondent.</td>
</tr>
<tr>
<td>1012</td>
<td>DEF dummy-long.</td>
</tr>
<tr>
<td></td>
<td>02 elem-identifier</td>
</tr>
<tr>
<td></td>
<td>02 dd-long-rtn</td>
</tr>
<tr>
<td></td>
<td>02 filler</td>
</tr>
<tr>
<td></td>
<td>TYPE BINARY 16</td>
</tr>
<tr>
<td></td>
<td>VALUE 1012.</td>
</tr>
<tr>
<td></td>
<td>TYPE BINARY 16</td>
</tr>
<tr>
<td></td>
<td>TYPE CHARACTER 80</td>
</tr>
<tr>
<td></td>
<td>VALUE SPACES.</td>
</tr>
<tr>
<td></td>
<td>FIELDS</td>
</tr>
<tr>
<td></td>
<td>• ELEM-IDENTIFIER is 1012.</td>
</tr>
<tr>
<td></td>
<td>• DD-LONG-RETN is OK.</td>
</tr>
<tr>
<td>1013</td>
<td>DEF data-depot-updated-by.</td>
</tr>
<tr>
<td></td>
<td>02 elem-identifier</td>
</tr>
<tr>
<td></td>
<td>02 dd-updated by-rtn</td>
</tr>
<tr>
<td></td>
<td>02 updated-by-corr-name</td>
</tr>
<tr>
<td></td>
<td>TYPE BINARY 16</td>
</tr>
<tr>
<td></td>
<td>VALUE 1013.</td>
</tr>
<tr>
<td></td>
<td>TYPE BINARY 16</td>
</tr>
<tr>
<td></td>
<td>PIC X(80).</td>
</tr>
</tbody>
</table>
Table 5-2. DDL Data Definitions for Profile File Data Elements with GET-PROFILE-ELEM Return Codes (Continued)

<table>
<thead>
<tr>
<th>Ident. No.</th>
<th>Definition and Field Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1013 FIELDS</td>
<td>cont'd</td>
</tr>
<tr>
<td>- ELEM-IDENTIFIER is 1013.</td>
<td></td>
</tr>
<tr>
<td>- DD-UPDATED-BY-RTN is one of the following:</td>
<td></td>
</tr>
<tr>
<td>0 OK</td>
<td></td>
</tr>
<tr>
<td>4238 E-TFER-PROFILE-ABSENT</td>
<td></td>
</tr>
<tr>
<td>- UPDATED-BY-CORR-NAME is the logged-on CORR-NAME of the correspondent who last issued an ALTER-PROFILE-ELEM UOW for this profile.</td>
<td></td>
</tr>
</tbody>
</table>

1014 DEF data-depot-device-option.

<table>
<thead>
<tr>
<th></th>
<th>TYPE</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>02 elem-identifier</td>
<td>BINARY 16</td>
<td>1014</td>
</tr>
<tr>
<td>02 ddd-option-rtn</td>
<td>BINARY 16</td>
<td></td>
</tr>
<tr>
<td>02 default-device-option</td>
<td>CHARACTER 50</td>
<td></td>
</tr>
<tr>
<td>02 filler</td>
<td>CHARACTER 30</td>
<td></td>
</tr>
<tr>
<td></td>
<td>VALUE SPACES</td>
<td></td>
</tr>
</tbody>
</table>

END.

FIELDS

- ELEM-IDENTIFIER is 1014.
- DDD-OPTION-RTN is one of the following:
  0 OK
  4239 E-MAIL-PROFILE-ABSENT
- DEFAULT-DEVICE-OPTION is used by the T/MAIL client to maintain additional parameters associated with the default device. The file name of the device is stored in the mail-output-filename field of Ident. No. 1003.
Table 5-2. DDL Data Definitions for Profile File Data Elements with GET-PROFILE-ELEM Return Codes (Continued)

<table>
<thead>
<tr>
<th>Ident. No.</th>
<th>Definition and Field Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1015 DEF data-depot-editor-name.</td>
<td></td>
</tr>
<tr>
<td>02 elem-identifier TYPE BINARY 16 VALUE 1015.</td>
<td></td>
</tr>
<tr>
<td>02 dde-filename-rtn TYPE BINARY 16.</td>
<td></td>
</tr>
<tr>
<td>02 mail-editor-name TYPE CHARACTER 36.</td>
<td></td>
</tr>
<tr>
<td>02 filler TYPE CHARACTER 44 VALUE SPACES.</td>
<td></td>
</tr>
<tr>
<td>END.</td>
<td></td>
</tr>
</tbody>
</table>

FIELDS

- ELEM-IDENTIFIER is 1015.
- DDE-FILENAME-RTN is one of the following:
  - 0 OK
  - 4239 E-MAIL-PROFILE-ABSENT
- MAIL-EDITOR-NAME is used by the T/MAIL client to maintain the program file name of the text editor to invoke on the correspondent's behalf.
Table 5-3. ALTER-PROFILE-ELEM Return Codes

<table>
<thead>
<tr>
<th>Ident. No.</th>
<th>Definition and Content</th>
<th>Return Codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DEF rtn-depot-password</td>
<td>TYPE BINARY 16.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 OK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4238 E-TFER-PROFILE-ABSENT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4300 E-PASSWORD-MATCH-FAIL</td>
</tr>
<tr>
<td>2</td>
<td>DEF rtn-depot-deltas</td>
<td>TYPE BINARY 16.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 OK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4072 E-UNITS-MUST-BE-DHM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4074 E-INVALID-REL-TIME-QTY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4238 E-TFER-PROFILE-ABSENT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4303 E-WINDOW-TOO-LARGE</td>
</tr>
<tr>
<td>3</td>
<td>DEF rtn-depot-priorities</td>
<td>TYPE BINARY 16.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 OK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4078 E-INVALID-PRIORITY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4093 E-SECURITY-VIOLATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4310 E-PRIORITY-SEQUENCE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4238 E-TFER-PROFILE-ABSENT</td>
</tr>
<tr>
<td>4</td>
<td>DEF rtn-depot-service-flags</td>
<td>TYPE BINARY 16.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 OK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4051 E-MUST-BE-YN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4052 E-RESERVED-MUST-BE-N</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4093 E-SECURITY-VIOLATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4238 E-TFER-PROFILE-ABSENT</td>
</tr>
<tr>
<td>5</td>
<td>DEF rtn-depot-log-flags</td>
<td>TYPE BINARY 16.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 OK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4051 E-MUST-BE-YN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4052 E-RESERVED-MUST-BE-N</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4238 E-TFER-PROFILE-ABSENT</td>
</tr>
</tbody>
</table>
Table 5-3. ALTER-PROFILE-ELEM Return Codes (Continued)

<table>
<thead>
<tr>
<th>Ident. No.</th>
<th>Definition and Content</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>DEF rtn-depot-privilege-flags</td>
<td>TYPE BINARY 16.</td>
</tr>
<tr>
<td></td>
<td>0 OK</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4093 E-SECURITY-VIOLATION</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4238 E-TFERPROFILE-ABSENT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4388 E-PRIV-MUST-BE-RWN</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>DEF rtn-depot-guardian-id</td>
<td>TYPE BINARY 16.</td>
</tr>
<tr>
<td></td>
<td>0 OK</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4238 E-TFERPROFILE-ABSENT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4304 E-INVALID-GUARDIAN-ID</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>DEF rtn-depot-session-ended</td>
<td>TYPE BINARY 16.</td>
</tr>
<tr>
<td></td>
<td>4302 E-NOT-UPDATABLE</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>DEF rtn-depot-created-time</td>
<td>TYPE BINARY 16.</td>
</tr>
<tr>
<td></td>
<td>4302 E-NOT-UPDATABLE</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>DEF rtn-depot-updated-time</td>
<td>TYPE BINARY 16.</td>
</tr>
<tr>
<td></td>
<td>4302 E-NOT-UPDATABLE</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>DEF rtn-sys-file-version</td>
<td>TYPE BINARY 16.</td>
</tr>
<tr>
<td></td>
<td>4302 E-NOT-UPDATABLE</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>DEF rtn-sys-window-limits</td>
<td>TYPE BINARY 16.</td>
</tr>
<tr>
<td></td>
<td>0 OK</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4072 E-UNITS-MUST-BE-DHM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4074 E-INVALID-REL-TIME-QTY</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4093 E-SECURITY-VIOLATION</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4241 E-TFER-CTL-REC-ABSENT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4303 E-WINDOW-TOO-LARGE</td>
<td></td>
</tr>
</tbody>
</table>
Table 5-3. ALTER-PROFILE-ELEM Return Codes (Continued)

<table>
<thead>
<tr>
<th>No.</th>
<th>Definition and Content</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>DEF rtn-depot-mail-print-ctl</td>
<td>TYPE BINARY 16.</td>
</tr>
<tr>
<td></td>
<td>0 OK</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4051 E-MUST-BE-YN</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4239 E-MAIL-PROFILE-ABSENT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4395 E-INVALID-INDENT</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>DEF rtn-depot-mail-flags</td>
<td>TYPE BINARY 16.</td>
</tr>
<tr>
<td></td>
<td>0 OK</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4239 E-MAIL-PROFILE-ABSENT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4051 E-MUST-BE-YN</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4382 E-MUST-BE-SLT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4383 E-MUST-BE-TCN</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4384 E-MUST-BE-MD</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>DEF rtn-depot-mail-volumes</td>
<td>TYPE BINARY 16.</td>
</tr>
<tr>
<td></td>
<td>0 OK</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4239 E-MAIL-PROFILE-ABSENT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4306 E-BAD-VOL-SUBVOL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4308 E-INVALID-NODE</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>DEF rtn-sys-mail-priority</td>
<td>TYPE BINARY 16.</td>
</tr>
<tr>
<td></td>
<td>0 OK</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4078 E-INVALID-PRIORITY</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4093 E-SECURITY-VIOLATION</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4242 E-MAIL-CTL-REC-ABSENT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4310 E-PRIORITY-SEQUENCE</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>DEF rtn-session-privileges</td>
<td>TYPE BINARY 16.</td>
</tr>
<tr>
<td></td>
<td>4302 E-NOT-UPDATABLE</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>DEF rtn-dummy-short</td>
<td>TYPE BINARY 16.</td>
</tr>
<tr>
<td></td>
<td>0 OK</td>
<td></td>
</tr>
<tr>
<td>Ident. No.</td>
<td>Definition and Content</td>
<td>Type</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>1003</td>
<td>DEF rtn-depot-mail-filename</td>
<td>TYPE BINARY 16.</td>
</tr>
<tr>
<td></td>
<td>0 OK</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4239 E-MAIL-PROFILE-ABSENT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4305 E-INVALID-FILENAME</td>
<td></td>
</tr>
<tr>
<td>1005</td>
<td>DEF rtn-depot-fax-routing</td>
<td>TYPE BINARY 16.</td>
</tr>
<tr>
<td></td>
<td>0 OK</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4240 E-FAX-PROFILE-ABSENT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5751 E-RECIP-NOT-FOUND</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5752 E-RECIP-BAD-NAME</td>
<td></td>
</tr>
<tr>
<td>1006</td>
<td>DEF rtn-depot-fax-folder1</td>
<td>TYPE BINARY 16.</td>
</tr>
<tr>
<td></td>
<td>0 OK</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4240 E-FAX-PROFILE-ABSENT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5676 E-FLD-NOT-FOUND</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5677 E-FLD-BAD-NAME</td>
<td></td>
</tr>
<tr>
<td>1011</td>
<td>DEF rtn-session-corr</td>
<td>TYPE BINARY 16.</td>
</tr>
<tr>
<td></td>
<td>4302 E-NOT-UPDATABLE</td>
<td></td>
</tr>
<tr>
<td>1012</td>
<td>DEF rtn-dummy-long</td>
<td>TYPE BINARY 16.</td>
</tr>
<tr>
<td></td>
<td>0 OK</td>
<td></td>
</tr>
<tr>
<td>1013</td>
<td>DEF rtn-depot-updated-by</td>
<td>TYPE BINARY 16.</td>
</tr>
<tr>
<td></td>
<td>4302 E-NOT-UPDATABLE</td>
<td></td>
</tr>
<tr>
<td>1014</td>
<td>DEF rtn-depot-device-option</td>
<td>TYPE BINARY 16.</td>
</tr>
<tr>
<td></td>
<td>0 OK</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4239 E-MAIL-PROFILE-ABSENT</td>
<td></td>
</tr>
</tbody>
</table>
Table 5-3. ALTER-PROFILE-ELEM Return Codes (Continued)

<table>
<thead>
<tr>
<th>Ident. No.</th>
<th>Definition and Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>1015</td>
<td>DEF rtn-depot-editor-name TYPE BINARY 16.</td>
</tr>
<tr>
<td></td>
<td>0   OK</td>
</tr>
<tr>
<td></td>
<td>4239 E-MAIL-PROFILE-ABSENT</td>
</tr>
<tr>
<td></td>
<td>4305 E-INVALID-FILENAME</td>
</tr>
<tr>
<td></td>
<td>4306 E-BAD-VOL-SUBVOL</td>
</tr>
<tr>
<td></td>
<td>4308 E-INVALID-NODE</td>
</tr>
<tr>
<td></td>
<td>4311 E-INVALID-EDITOR</td>
</tr>
</tbody>
</table>
GET-RECIP-REC retrieves recipient records for a package.

DEF get-recip-rec-uow.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 126.
  02 recip-key.
    03 item-id PIC X(12).
    04 dummy.
    03 recip-name PIC X(120).
  02 options.
    03 get-exact.
    03 generic-suffix.
    03 skip-exact.
    03 strip-local-node-name.
    03 reserved-4.
    03 reserved-5.
    03 reserved-6.
    03 filter-by-recip-type.
    03 filter-by-status-flag.
      04 delivered.
      04 byte.
      04 examined.
      04 certification-acked.
      04 canceled.
      04 expired.
      04 async-resolved.
      04 reserved-6.
      04 reserved-7.
      04 is-original-recip.
      04 is-local-dlist.
      04 is-local-corr.
      04 is-remote-recip.
      04 reserved-12.
      04 reserved-13.
      04 reserved-14.
      04 reserved-15.
  02 filter-values.
    03 recip-type.
    03 status-flags.
      04 delivered.
      04 byte.
      04 examined.
      04 certification-acked.
      04 canceled.
      04 expired.

TYPE UOW-HDR.
PIC X(l2).
PIC X(l20).
TYPE BOOLEAN.
REDEFINES DELIVERED PIC X.
TYPE BOOLEAN.
TYPE BOOLEAN.
TYPE BOOLEAN.
TYPE BOOLEAN.
TYPE BOOLEAN.
TYPE BOOLEAN.
TYPE BOOLEAN.
TYPE BOOLEAN.
TYPE BOOLEAN.
TYPE BOOLEAN.
TYPE BOOLEAN.
TYPE BOOLEAN.
TYPE BOOLEAN.
TYPE BOOLEAN.
TYPE BOOLEAN.
TYPE BOOLEAN.
TYPE BOOLEAN.
TYPE BOOLEAN.
TYPE BOOLEAN.
TYPE BOOLEAN.
TYPE BOOLEAN.
PIC 9(4) COMP.
UOW Descriptions
GET-RECIP-REC

04 async-resolved TYPE BOOLEAN.
04 reserved-6 TYPE BOOLEAN VALUE "N".
04 reserved-7 TYPE BOOLEAN.
04 is-original-recip TYPE BOOLEAN.
04 is-local-dlist TYPE BOOLEAN.
04 is-local-corr TYPE BOOLEAN.
04 is-remote-recip TYPE BOOLEAN.
04 reserved-12 TYPE BOOLEAN VALUE "N".
04 reserved-13 TYPE BOOLEAN VALUE "N".
04 reserved-14 TYPE BOOLEAN VALUE "N".
04 reserved-15 TYPE BOOLEAN VALUE "N".
02 num-requested TYPE BINARY 16 UNSIGNED VALUE 10.

END.

DEF get-recip-rec-rsp.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 126.
  02 retn-code TYPE BINARY 16.
  02 retn-code-detail TYPE BINARY 16.
  02 num-returned TYPE BINARY 16.
  02 recip-name TYPE BINARY 16.
  02 recip-type TYPE BINARY 16.
  02 deliv-err TYPE BINARY 16.
  03 deliv-status-flags.
    04 delivered TYPE BOOLEAN.
    04 byte TYPE BOOLEAN.
    04 examined TYPE BOOLEAN.
    04 certification-acked TYPE BOOLEAN.
    04 canceled TYPE BOOLEAN.
    04 expired TYPE BOOLEAN.
    04 async-resolved TYPE BOOLEAN.
    04 reserved-6 TYPE BOOLEAN VALUE "N".
    04 reserved-7 TYPE BOOLEAN VALUE "N".
    04 is-original-recip TYPE BOOLEAN.
    04 is-local-dlist TYPE BOOLEAN.
    04 is-local-corr TYPE BOOLEAN.
    04 is-remote-recip TYPE BOOLEAN.
    04 reserved-12 TYPE BOOLEAN.
    04 reserved-13 TYPE BOOLEAN.

5-145
UOW Descriptions
GET-RECIP-REC

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>04 reserved-14</td>
<td>TYPE BOOLEAN VALUE &quot;N&quot;.</td>
<td></td>
</tr>
<tr>
<td>04 reserved-15</td>
<td>TYPE BOOLEAN VALUE &quot;N&quot;.</td>
<td></td>
</tr>
<tr>
<td>03 date-delivered.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>04 year</td>
<td>PIC 9(4).</td>
<td></td>
</tr>
<tr>
<td>04 month</td>
<td>PIC 9(2).</td>
<td></td>
</tr>
<tr>
<td>04 day-of-month</td>
<td>PIC 9(2).</td>
<td></td>
</tr>
<tr>
<td>04 hour</td>
<td>PIC 9(2).</td>
<td></td>
</tr>
<tr>
<td>04 minute</td>
<td>PIC 9(2).</td>
<td></td>
</tr>
<tr>
<td>04 second</td>
<td>PIC 9(2).</td>
<td></td>
</tr>
</tbody>
</table>

END.

GET-RECIP-REC FIELDS. The fields defined in this UOW are:

- HDR is the UOW header. The UOW-CODE value is 126.
- RECIP-KEY identifies the item ID of the package header to which the recipient list applies, and the name of the recipient at which the retrieval should begin in the recipient list.
- OPTIONS establishes criteria for selectively retrieving records. Six options are provided. In these options, the entry Y selects the option, while the entry N suppresses it.
  - GET-EXACT retrieves the recipient record that exactly matches the name specified in RECIP-KEY.
  - GENERIC-SUFFIX retrieves all recipient records for the recipient specified in RECIP-KEY, regardless of whatever suffix might be appended to the recipient name. If GET-EXACT is set to Y, GENERIC-SUFFIX is ignored.
  - SKIP-EXACT indicates whether TISERV begins its retrieval with the record specified in RECIP-KEY, or with the next record in the recipient file.
    - Y = start retrieval with the first record following the one specified in RECIP-KEY, whether or not the specified record actually exists.
    - N = start retrieval with the exact record specified in RECIP-KEY (if that record is present), or with the first record following that one (if the exact record is not present).
STRIP-LOCAL-NODE-NAME omits the node name from the returned recipient names if the recipients are local. Remote node names are always included, regardless of the setting of this field.

RESERVED-4 through RESERVED-6 are reserved for use by Tandem; these fields must be set to N.

FILTER-BY-RECIP-TYPE selects only those records identified by the value of the RECIP-TYPE field in FILTER-VALUES, beginning with the record identified by RECIP-KEY.

FILTER-BY-STATUS-FLAG selects the status flags by which you wish to filter the records for retrieval. The status flags are explained in the STATUS-FLAGS field of FILTER-VALUES.

FILTER-VALUES defines the filter by which records are selected for retrieval. This field applies only if FILTER-BY-RECIP-TYPE or FILTER-BY-STATUS-FLAG is selected in the OPTIONS field. FILTER-VALUES includes the following:

RECIP-TYPE specifies a recipient type and determines that only records of this type will be retrieved. FILTER-BY-RECIP-TYPE must be set to Y.

STATUS-FLAGS specifies that only recipient records whose status flag matches this input value will be retrieved. FILTER-BY-STATUS-FLAG must be set to Y.

The status flags maintained by TRANSFER contain status information about the delivery of a package with respect to a particular recipient. These flags are set by the TAREQ that delivers the package to the recipient.

The status flags denote the following information:

DELIVERED indicates whether the package was delivered to the recipient by the TAREQ.

EXAMINED indicates whether the recipient issued the ACK-RECEIPT UOW for the package.

CERTIFICATION-ACKED indicates whether the TAREQ transmitted a certification package to the sender.

CANCELED indicates whether delivery of the package to this recipient was canceled by the sender.

EXPIRED indicates whether the package has been removed from the recipient INBOX because of expiration.
ASYNC-RESOLVED indicates for a distribution list whether the list has been expanded by the TAREQ; or indicates for a remote recipient whether the package has been transported by the TAREQ.

RESERVED-6 through RESERVED-15 are reserved for use by Tandem; these fields are always set to N.

IS-ORIGINAL-RECIP indicates whether this recipient was an original recipient of the package, or whether the recipient name was derived from a distribution list.

IS-LOCAL-DLIST indicates whether the recipient is a local distribution list.

IS-LOCAL-CORR indicates whether the recipient is a local correspondent.

IS-REMOTE-RECIP indicates whether the recipient is defined at a local or remote node.

NOTE

Except for the DELIVERED, IS-ORIGINAL-RECIP, and EXAMINED flags, the internal status flags contain information about rapidly changing events; they might not always provide an up-to-the-minute reflection of actual status.

• NUM-REQUESTED is the number of records to be retrieved. The value directly influences the size of the response.

• RETN-CODE is the return code. TISERV returns a code in this field to indicate one of the following entries:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OK</td>
</tr>
<tr>
<td>4001</td>
<td>W-EOF</td>
</tr>
<tr>
<td>4035</td>
<td>E-ITEM-NOT-FOUND</td>
</tr>
<tr>
<td>4042</td>
<td>E-ITEM-NOT-PKG-HDR</td>
</tr>
<tr>
<td>4051</td>
<td>E-MUST-BE-YN</td>
</tr>
<tr>
<td>4052</td>
<td>E-RESERVED-MUST-BE-N</td>
</tr>
<tr>
<td>4092</td>
<td>E-INVALID-NUM-RQSTD</td>
</tr>
<tr>
<td>5752</td>
<td>E-RECIP-BAD-NAME</td>
</tr>
<tr>
<td>5774</td>
<td>E-RECIP-BAD-SUFFIX</td>
</tr>
</tbody>
</table>

• RETN-CODE DETAIL is an error number returned by a subsystem other than TRANSFER, or is a further qualification of an error detected by TRANSFER.

• NUM-RETURNED is the number of records returned in the response.
RECIPS-RETURNED contains the records returned, and includes for each record fields that indicate:

- the recipient name and type (RECIP-NAME and RECIP-TYPE)
- delivery error and status flags set by the TAREQs that delivered the package (DELIV-ERR and DELIV-STATUS-FLAGS)
- the time and date at the recipient node when the package was delivered (DATE-DELIVERED).

GET-RECIP-REC OPERATION. GET-RECIP-REC retrieves local recipient records for the package identified by RECIP-KEY. The records are placed in the array named RECIPS-RETURNED in ascending order by system name.

The retrieval can begin with the exact recipient name specified in RECIP-KEY if such a recipient name exists; or with the next one depending on the value of the SKIP-EXACT field in the OPTIONS structure.

To reference the first recipient in the recipient list, specify an empty RECIP-NAME field:

- RECIP-NAME = all blanks, or begin the field with a null character.

To retrieve a recipient list beginning with the first recipient in the list, set

- SKIP-EXACT = Y
- GET-EXACT = N
- GENERIC-SUFFIX = N

When you specify filtering criteria in OPTIONS and FILTER-VALUES, only recipient records that meet all of these criteria are returned.

The number of recipient records retrieved might be less than the number requested if there are fewer recipients than meet the request criteria. In this case, the warning W-EOF is returned in the RETN-CODE field.
The recipients for the local copy of any package include:

- all recipients explicitly added by a client, regardless of the nodes at which the recipients are registered
- those local recipients that were derived from distribution lists, if TRANSFER has already attempted delivery to the recipient
- possibly some, but not necessarily all, local recipients that were derived from distribution lists, if TRANSFER has not yet attempted delivery to those recipients
- possibly some, but not necessarily all, remote recipients that were derived from distribution lists, whether or not TRANSFER has already attempted to deliver the package to those recipients.

Local records concerning remote recipients do not necessarily contain up-to-date status information; therefore, you should obtain status information for remote recipients from the corresponding remote copy of TISERV. To obtain status information, do the following:

1. Send a pair of GET-CONFIG-NAME UOWs to your local TISERV to retrieve the names of the PATHMON and TISERV on the remote system.
2. Start a session with the remote TISERV (you must have a depot on the remote system).
3. Send the GET-RECIP-REC request to the remote system.
NOOP (UOW Code 0)

NOOP transmits a dummy UOW to TISERV. This UOW does not require the establishment of a session.

```
DEF noop-uow.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 0.
  02 retn-code TYPE BINARY 16.
  02 retn-code-detail TYPE BINARY 16.
  02 data-size TYPE BINARY 16 UNSIGNED.
  02 noop-data PIC X OCCURS 0 TO !N! 1000 TIMES DEPENDING ON data-size.
END.

DEF noop-rsp.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 0.
  02 retn-code TYPE BINARY 16.
  02 retn-code-detail TYPE BINARY 16.
  02 data-size TYPE BINARY 16 UNSIGNED.
  02 noop-data PIC X OCCURS 0 TO !N! 1000 TIMES DEPENDING ON data-size.
END.
```

NOOP FIELDS. The fields defined in this UOW are:

- HDR is the UOW header. The UOW-CODE value is 0.
- RETN-CODE contains a zero.
- RETN-CODE-DETAIL contains a zero.
- DATA-SIZE is the length of the data transmitted and returned, specified in bytes.
- NOOP-DATA is an array that contains the data itself. In the OCCURS clause of the DDL definition for this structure, N is any value less than 2000.
NOTE

If the value of DATA-SIZE is an odd number and additional UOWs follow this one, append a one-byte FILLER after the NOOP-DATA structure.

NOOP OPERATION. NOOP lets you transmit a dummy UOW to TISERV and receive a dummy response. The data transmitted in the UOW is returned without modification in the response.

To help identify all IPC messages that are logged, you can use the NOOP UOW to include, after every IPC header, client data that identifies the source of the transaction.

As an example, you can replace the NOOP-DATA structure with the following DDL definition; upon receipt by TISERV, this optional data will be echoed back to your application:

```
02 client-info.
   03 application PIC X !((N))!.
   03 module PIC X !((N))!.
   03 func-code PIC X !((N))!.
   03 term-id PIC X (15).
   03 filler PIC X.
```

This definition provides the following tracking information:

APPLICATION contains information that helps identify your application.

MODULE is the SCREEN COBOL program unit code name.

FUNC-CODE is the related transaction (function key) code.

TERM-ID is the terminal ID of the correspondent.
READ-NEXT-MEMBER (UOW Code 219)

READ-NEXT-MEMBER retrieves members of a distribution list.

```
DEF read-next-member-uow.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 219.
  02 filler TYPE BINARY 16.
  02 filler TYPE BINARY 16.
  02 corr-name TYPE PIC X(80) VALUE SPACES.
  02 dlist-name TYPE PIC X(80).
  02 start-member TYPE PIC X(120) VALUE SPACES.
  02 start-type TYPE CHARACTER 1 VALUE "A".
  02 filler TYPE BOOLEAN.
  02 name-format TYPE BINARY 16 VALUE 1.
  02 num-wanted TYPE BINARY 16 UNSIGNED.
  02 filler TYPE BINARY 16 UNSIGNED VALUE 0.
END.
```

```
DEF read-next-member-rsp.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 219.
  02 retn-code TYPE BINARY 16.
  02 retn-code-detail TYPE BINARY 16.
  02 corr-name TYPE PIC X(80) VALUE SPACES.
  02 dlist-name TYPE PIC X(80).
  02 start-member TYPE PIC X(120) VALUE SPACES.
  02 start-type TYPE CHARACTER 1 VALUE "A".
  02 update-flag TYPE BOOLEAN.
  02 name-format TYPE BINARY 16 VALUE 1.
  02 num-wanted TYPE BINARY 16 UNSIGNED VALUE 14.
  02 num-returned TYPE BINARY 16 UNSIGNED VALUE 0.
  02 member-name OCCURS 0 TO 14 TIMES DEPENDING ON num-returned PIC X(120).
END.
```

READ-NEXT-MEMBER FIELDS. The fields defined in this UOW are:

- HDR is the UOW header. The UOW-CODE value is 219.
UOW Descriptions
READ-NEXT-MEMBER

• CORR-NAME is the name of the correspondent represented by the calling process. Only system administrators can enter correspondent names other than their own in this field; correspondents without system administrator privileges must enter their own names or leave the field blank. On input, you can use a partially qualified name, including wildcard characters. In the response, the fully qualified name is returned.

• DLIST-NAME is the name of the distribution list from which the members are retrieved. This can be a partially qualified name, including wildcard characters. In the response, the fully qualified name is returned.

• START-MEMBER is the distribution list member with which the retrieval should start. To start at the beginning of the distribution list, set this field to spaces or null characters. In the response, the fully qualified name to be transmitted in the next call to this UOW is returned.

• START-TYPE indicates whether the retrieval should begin with the exact record specified in START-MEMBER, or with the next record following that record.

   E (START-EXACT) = the exact record

   A (START-AFTER) = the next record

   If the request is successful, TRANSFER returns the value A in the response, which is appropriate for the next request to continue getting members from the distribution list.

• NAME-FORMAT indicates the format in which the name is to be returned, as follows:

<table>
<thead>
<tr>
<th>Setting</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>NODE-OPTIONAL 0</td>
<td>The name is returned in system-optimal format, which omits the node designation if the name is defined at the same node as the logged-on correspondent.</td>
</tr>
<tr>
<td>FULL-QUALIFICATION 1</td>
<td>The name is returned in fully qualified format, which is the TRANSFER name in its full form: name[.name] @system</td>
</tr>
</tbody>
</table>
### Setting Meaning

<table>
<thead>
<tr>
<th>Setting</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRE-SEARCH-LIST 2</td>
<td>Names belonging to CORR-NAME are returned as simple names; for example, the name SYS-ADMIN.DLIST @TM is returned simply as DLIST if the correspondent requesting the name is SYS-ADMIN @TM. Names belonging to other correspondents at this node are returned as name[.name]. Names belonging to correspondents at other nodes are returned as fully qualified.</td>
</tr>
</tbody>
</table>

- **NUM-WANTED** is the number of members requested. NUM-WANTED and the number of occurrences of MEMBER-NAME are limited only by the maximum size of the IPC in your TRANSFER configuration. In the response, the input to this field is echoed as output.

- **RETN-CODE** is the return code. TISERV returns a code in this field to indicate one of the following entries.

To indicate successful retrieval:

0 OK

To indicate problems with the correspondent name:

5600 E-CORR-NSRV-ERR
5601 E-CORR-NOT-FOUND
5602 E-CORR-BAD-NAME
5604 E-CORR-NO-SUCH-NODE
5606 E-CORR-NSRV-NOT-FOUND
5607 E-CORR-NSRV-DOWN
5611 E-CORR-NET-DOWN
5622 E-CORR-NOT-SAME-NODE
5623 E-CORR-AMBIGUOUS-NAME

To indicate problems with the distribution list name:

5625 E-DLIST-NSRV-ERR
5626 E-DLIST-NOT-FOUND
5627 E-DLIST-BAD-NAME
5629 E-DLIST-NO-SUCH-NODE
5631 E-DLIST-NSRV-NOT-FOUND
5632 E-DLIST-NSRV-DOWN
5636 E-DLIST-NET-DOWN
5647 E-DLIST-NOT-SAME-NODE
5648 E-DLIST-AMBIGUOUS-NAME

To indicate other problems:

4001 W-EOF
4010 E-BAD-TRANSACTION
4093 E-SECURITY-VIOLATION
4201 E-CONTEXT-ERR
4229 E-MUST-BE-EA
4902 E-ERR-PROFILE-FILE
4914 E-ERR-DLIST-FILE
UOW Descriptions
READ-NEXT-MEMBER

• RETN-CODE-DETAIL is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

• UPDATE-FLAG indicates whether an update of the distribution list is permitted by the logged-on correspondent:
  \[ Y \text{ (UPDATE-ALLOWED)} = \text{updating is allowed} \]
  \[ N \text{ (UPDATE-NOT-ALLOWED)} = \text{updating is not allowed} \]

• NUM-RETURNED is the number of members actually retrieved.

• MEMBER-NAME is an array containing the distribution list member names retrieved.

READ-NEXT-MEMBER OPERATION. The READ-NEXT-MEMBER UOW retrieves from the distribution list identified by DLIST-NAME, the names of members in the range identified by START-MEMBER and NUM-WANTED. To begin a retrieval with the first member on the list, set the START-MEMBER field to spaces. The names of the retrieved members are stored in the array MEMBER-NAME.

The response is constructed so that it can also act as a request. To use a response in this way, set NUM-RETURNED to zero and retransmit the UOW. The START-MEMBER and START-TYPE fields are set up to cause proper action on the next call; thus, you need not alter these fields after the first call. Because of this two-way feature, those fields that contain only return values are specified as FILLER fields in the request.

If the number of elements remaining in the list is less than or equal to the number requested, or if no names were found, the warning W-EOF is returned in the RETN-CODE field.
READ-NEXT-NAME (UOW Code 224)

READ-NEXT-NAME retrieves correspondent, distribution list, or folder names defined in TRANSFER.

```
DEF read-next-name-uow.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 224.
  02 filler TYPE BINARY 16.
  02 filler TYPE BINARY 16.
  02 corr-name PIC X(80) VALUE SPACES.
  02 partial PIC X(80) VALUE SPACES.
  02 name-type PIC X(32).
  02 kind TYPE BINARY 16.
  02 restriction-flag TYPE BINARY 16 VALUE 2.
  02 alias-action-flag TYPE BINARY 16 VALUE 2.
  02 name-format TYPE BINARY 16 VALUE 1.
  02 num-wanted TYPE BINARY 16 UNSIGNED.
  02 matcher PIC X(80).
  02 dir-no TYPE BINARY 16.
  02 filler TYPE BINARY 16 UNSIGNED VALUE 0.
END.

DEF read-next-name-rsp.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 224.
  02 retn-code TYPE BINARY 16.
  02 retn-code-detail TYPE BINARY 16.
  02 corr-name PIC X(80) VALUE SPACES.
  02 partial PIC X(80) VALUE SPACES.
  02 name-type PIC X(32).
  02 kind TYPE BINARY 16.
  02 restriction-flag TYPE BINARY 16 VALUE 2.
  02 alias-action-flag TYPE BINARY 16 VALUE 2.
  02 name-format TYPE BINARY 16 VALUE 1.
  02 num-wanted TYPE BINARY 16 UNSIGNED.
  02 matcher PIC X(80).
  02 dir-no TYPE BINARY 16.
  02 num-returned TYPE BINARY 16 UNSIGNED VALUE 0.
```
READ-NEXT-NAME FIELDS. The fields defined in this UOW are:

- **HDR** is the UOW header. The UOW-CODE value is 224.

- **CORR-NAME** is the name of the correspondent represented by the calling process. Only system administrators can enter correspondent names other than their own in this field; correspondents without system administrator privileges must enter their own names or leave the field blank. On input, you can use a partially qualified name, including wildcard characters. In the response, the fully qualified name is returned.

Only system administrators can retrieve folder names for depots other than their own; to do this, they must enter in CORR-NAME the name of the correspondent who owns the folders. Anyone can retrieve names for distribution lists or correspondents defined at any depot.

- **PARTIAL** is the partially qualified name or pattern of the names to be retrieved from the TRANSFER name directory. This name can include wildcard characters. In the response, the input to this field is echoed as output.

- **NAME-TYPE** is the type of object for which the names are to be retrieved. This can be:
  
  CORR (for correspondent)  DLIST (for distribution list)
  FOLDER (for folder)  blank (for any type)

  In the response, the input to this field is echoed as output.

- **KIND** is the kind of search used for the names, as follows.
### UOW Descriptions

**READ-NEXT-NAME**

<table>
<thead>
<tr>
<th>Setting</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIRST-CALL (0)</td>
<td>Searches for the first names encountered that match the contents of the PARTIAL field. In the response, this value is converted to RECALL (2) if at least one name is returned.</td>
</tr>
<tr>
<td>RECALL (2)</td>
<td>Continues the previous search from the point specified by the contents of MATCHER and DIR-NO, which were returned on a previous call.</td>
</tr>
<tr>
<td>SKIP (3)</td>
<td>Reserved for use by Tandem.</td>
</tr>
</tbody>
</table>

- **RESTRICTION-FLAG** restricts the name search. Settings are as follows:

<table>
<thead>
<tr>
<th>Setting</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESTRICT-TO-WDIR (0)</td>
<td>Restrict the search to names belonging to the CORR-NAME depot.</td>
</tr>
<tr>
<td>PART-RESTRICT-TO-WDIR (1)</td>
<td>If PARTIAL contains a period or a node name, first search the CORR-NAME depot, and then search the entire name directory; this latter search will only find objects if PARTIAL includes a correspondent name, which might include wildcard characters. If PARTIAL does not contain a period and is not fully qualified, restrict the search to the CORR-NAME depot.</td>
</tr>
<tr>
<td>UNRESTRICTED-DIR (2)</td>
<td>First, search the CORR-NAME depot; then search the entire name directory. This latter search will only find objects if PARTIAL includes a correspondent name, which might include wildcard characters.</td>
</tr>
</tbody>
</table>

In the response, the input to this field is echoed as output.

- **ALIAS-ACTION-FLAG** is reserved for use by Tandem.

In the response, the input to this field is echoed as output.
UOW Descriptions
READ-NEXT-NAME

- **NAME-FORMAT** indicates the format in which the names are to be returned, as follows:

<table>
<thead>
<tr>
<th>Setting</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>NODE-OPTIONAL (0)</td>
<td>The name is returned in system-optional format, which omits the node designation if a name is defined at the same node as the logged-on correspondent.</td>
</tr>
<tr>
<td>FULL-QUALIFICATION (1)</td>
<td>The name is returned in full qualification format, which returns the TRANSFER name in its full form: name[.name] @system</td>
</tr>
<tr>
<td>PRE-SEARCH-LIST (2)</td>
<td>Names belonging to CORR.NAME are returned as simple names. For example, the name SYS-ADMIN.FOLDER1 @TM is returned simply as FOLDER1 if the correspondent requesting the name is SYS-ADMIN @TM. Names belonging to other correspondents at the node are returned as name[.name]. Names belonging to correspondents at other nodes are returned as fully qualified.</td>
</tr>
</tbody>
</table>

The default value for this field is 1. In the response, the input to this field is echoed as output.

- **NUM-WANTED** is the number of names to be retrieved. In the response, the input to this field is echoed as output. NUM-WANTED and the NAME-GROUP array size are limited only by the maximum size of the IPC in your TRANSFER configuration.

- **MATCHER** and **DIR-NO** are internal restart pointers that are returned in the first call with this UOW; they must be used in subsequent READ-NEXT-NAME requests where the KIND field is set to RECALL (2). These values are ignored if the KIND field is set to FIRST-CALL (0).

- **RETN-CODE** is the return code. TISERV returns a code in this field to indicate one of the following entries.

  To indicate a successful retrieval:

  0  OK

  To indicate problems with the correspondent name or with names that are designated as any type.
UOW Descriptions

READ-NEXT-NAME

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5600</td>
<td>E-CORR-NSRV-ERR</td>
</tr>
<tr>
<td>5601</td>
<td>E-CORR-NOT-FOUND</td>
</tr>
<tr>
<td>5602</td>
<td>E-CORR-BAD-NAME</td>
</tr>
<tr>
<td>5603</td>
<td>E-CORR-BAD-TYPE</td>
</tr>
<tr>
<td>5604</td>
<td>E-CORR-NO-SUCH-NODE</td>
</tr>
<tr>
<td>5605</td>
<td>E-CORR-NSRV-NOT-FOUND</td>
</tr>
<tr>
<td>5606</td>
<td>E-CORR-NSRV-DOWN</td>
</tr>
<tr>
<td>5607</td>
<td>E-CORR-NET-DOWN</td>
</tr>
<tr>
<td>5611</td>
<td>E-CORR-NET-DOWN</td>
</tr>
<tr>
<td>5622</td>
<td>E-CORR-NOT-SAME-NODE</td>
</tr>
<tr>
<td>5623</td>
<td>E-CORR-AMBIGUOUS-NAME</td>
</tr>
</tbody>
</table>

To indicate problems with other names, similar error messages are returned. These depend on the contents of the NAME-TYPE field.

To indicate other problems:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4001</td>
<td>W-EOF</td>
</tr>
<tr>
<td>4010</td>
<td>E-BAD-TRANSACTION</td>
</tr>
<tr>
<td>4201</td>
<td>E-CONTEXT-ERR</td>
</tr>
<tr>
<td>4202</td>
<td>E-ERR-PROFILE-FILE</td>
</tr>
</tbody>
</table>

- RETN-CODE-DETAIL is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.
- NUM-RETURNED is the number of names returned.
- NAME-GROUP is an array that contains the name returned in the NAME field. You can specify any number of occurrences, restricted only by the length of the entire IPC that contains this UOW.

ALIAS-TYPE has no significance in this version of TRANSFER.

READ-NEXT-NAME OPERATION. READ-NEXT-NAME scans the TRANSFER name directory and returns the next name or group of names encountered on each call. You can use this UOW to display lists of folder names, distribution list names, or correspondent names. In selecting names for retrieval, this UOW supports wildcard matching of names.

If the number of elements remaining in the list is less than or equal to the number requested, or if no elements matching the searching criteria were found, the warning W-EOF is returned in the RETN-CODE field.

In SCREEN COBOL programs, it is recommended that the response definition be used as the definition for the UOW request. You accomplish this by setting the NUM-RETURNED field to zero before transmitting each request. This ensures that the length of the response is the same as the length of the UOW.

If a name returned by this UOW is given back to TRANSFER, only the fully qualified form of this name is guaranteed to refer to the same object. If any other NAME-FORMAT is specified, the name might cause a different object to be identified.
UOW Descriptions
READ-PROFILE-REC

READ-PROFILE-REC (UOW Code 212)

READ-PROFILE-REC reads a user-maintained depot profile record. This UOW is maintained for version compatibility only. READ-PROF-REC-A02 is the recommended UOW.

DEF read-profile-rec-uow.
  02 hdr.
   03 self-ident PIC AA VALUE "UW".
   03 uow-code TYPE BINARY 16 UNSIGNED VALUE 212.
  02 corr-name PIC X(80) VALUE SPACES.
  02 rec-type PIC 9(4) COMP.
  02 rec-seq-num PIC 9(4) COMP.
  02 depot-flag TYPE BOOLEAN VALUE "Y".
  02 filler TYPE CHARACTER 1.
  02 approximate-flag TYPE BINARY 16.
END.

DEF read-profile-rec-rsp.
  02 hdr.
   03 self-ident PIC AA VALUE "UW".
   03 uow-code TYPE BINARY 16 UNSIGNED VALUE 212.
  02 retn-code TYPE BINARY 16.
  02 retn-code-detail TYPE BINARY 16.
  02 corr-name PIC X(80).
  02 rec-type PIC 9(4) COMP.
  02 rec-seq-num PIC 9(4) COMP.
  02 update-control PIC S9(4) COMP.
  02 num-data TYPE BINARY 16 UNSIGNED.
  02 profile-data.
     03 byte PIC X OCCURS 0 TO 1000 TIMES DEPENDING ON num-data.
END.

READ-PROFILE-REC FIELDS. The fields defined in this UOW are:
- HDR is the UOW header. The UOW-CODE value is 212.
• CORR-NAME is the name of the correspondent whose depot profile record is to be read. This is also the name of the correspondent represented by the calling process. Only system administrators can enter correspondent names other than their own in this field; correspondents without system administrator privileges must enter their own names or leave the field blank. On input, you can use a partially qualified name, including wildcard characters. In the response, the fully qualified name is returned.

• REC-TYPE is the type assigned to the record by your application. In the response, the type of the record accessed for retrieval is returned.

• REC-SEQ-NUM is the sequence number assigned to the record. In the response, the sequence number of the record retrieved is returned.

• DEPOT-FLAG determines whether the UOW references a depot profile record or a system control record.

  Y = a depot profile record, as indicated by CORR-NAME

  N = a system control record

  If you make no entry in this field, Y is assumed.

• APPROXIMATE-FLAG is the type of record positioning preferred, as follows:

  Setting          Meaning
  
  EXACT-POS (2)   Requests exact record positioning, in which TISERV reads the exact user profile record specified by REC-TYPE and REC-SEQ-NUM.

  APPROXIMATE-POS (0) Requests approximate record positioning, in which TISERV reads the next user profile record at or past the record specified by REC-TYPE and REC-SEQ-NUM. This option does not permit reading beyond the end of the depot.

• RETN-CODE is the return code. TISERV returns a code in this field to indicate one of the following entries.

  To indicate successful record retrieval:

    0  OK

  To indicate problems with the correspondent name:
UOW Descriptions

READ-PROFILE-REC

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5600</td>
<td>E-CORR-NSRV-ERR</td>
</tr>
<tr>
<td>5601</td>
<td>E-CORR-NOT-FOUND</td>
</tr>
<tr>
<td>5602</td>
<td>E-CORR-BAD-NAME</td>
</tr>
<tr>
<td>5603</td>
<td>E-CORR-BAD-TYPE</td>
</tr>
<tr>
<td>5606</td>
<td>E-CORR-NSRV-NOT-FOUND</td>
</tr>
<tr>
<td>5607</td>
<td>E-CORR-NSRV-DOWN</td>
</tr>
<tr>
<td>5611</td>
<td>E-CORR-NET-DOWN</td>
</tr>
<tr>
<td>5622</td>
<td>E-CORR-NOT-SAME-NODE</td>
</tr>
<tr>
<td>5623</td>
<td>E-CORR-AMBIGUOUS-NAME</td>
</tr>
<tr>
<td>5625</td>
<td>E-CORR-AMBIGUOUS-NAME</td>
</tr>
<tr>
<td>4001</td>
<td>W-EOF</td>
</tr>
<tr>
<td>4010</td>
<td>E-BAD-transaction</td>
</tr>
<tr>
<td>4046</td>
<td>E-INVALID-REC-TYPE</td>
</tr>
<tr>
<td>4051</td>
<td>E-MUST-BE-YN</td>
</tr>
<tr>
<td>4093</td>
<td>E-SECURITY-VIOLATION</td>
</tr>
<tr>
<td>4201</td>
<td>E-CONTEXT-ERR</td>
</tr>
<tr>
<td>4220</td>
<td>W-EXACT-REC-READ</td>
</tr>
<tr>
<td>4221</td>
<td>W-NEXT-REC-READ</td>
</tr>
<tr>
<td>4902</td>
<td>E-ERR-PROFILE-FILE</td>
</tr>
</tbody>
</table>

To indicate other problems:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4201</td>
<td>E-CONTEXT-ERR</td>
</tr>
<tr>
<td>4220</td>
<td>W-EXACT-REC-READ</td>
</tr>
<tr>
<td>4221</td>
<td>W-NEXT-REC-READ</td>
</tr>
<tr>
<td>4902</td>
<td>E-ERR-PROFILE-FILE</td>
</tr>
</tbody>
</table>

• RETN-CODE-DETAIL is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

• UPDATE-CONTROL is the counter used to control concurrent updates of the profile record. You must pass this key to the WRITE-PROFILE-REC to assure that no other process has changed the record between the read and write operations.

• NUM-DATA is the number of bytes read by TISERV.

• PROFILE-DATA is the array that contains the record read by TISERV. You can modify the array size within the system limitation of 1500 bytes.

READ-PROFILE-REC OPERATION. READ-PROFILE-REC reads a user-maintained record from the depot Profile file and returns the data portion of the record. Although the profile record header is not returned, the record type and sequence number from this header are made available as separate fields (REC-TYPE and REC-SEQ-NUM).

The UOW must start on a word boundary. Each response UOW consumes a whole number of words of the IPC reply; thus, TRANSFER automatically includes a padding byte if the returned record is an odd number of bytes long.

The record is read from the depot (correspondent) identified by CORR-NAME, and is selected in accordance with REC-TYPE, REC-SEQ-NUM, and APPROXIMATE-FLAG. If the DEPOT-FLAG was set to indicate a system control record, the single copy of this record in the system was read. The record is returned to the array named PROFILE-DATA.

If approximate positioning is requested, a warning is returned specifying whether or not the exactly matching record was found. The W-EOF warning is returned if the requested record does not exist.
READ-PROF-REC-A02 (UOW Code 231)

READ-PROF-REC-A02 reads a user-maintained depot profile record.

```
DEF read-prof-rec-a02-uow.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 231.
  02 corr-name PIC X(80) VALUE SPACES.
  02 rec-type PIC 9(4) COMP.
  02 rec-seq-num PIC 9(4) COMP.
  02 options.
    03 skip-exact TYPE BOOLEAN.
    03 any-rec-type TYPE BOOLEAN.
    03 any-seq-num TYPE BOOLEAN.
    03 depot-flag TYPE BOOLEAN.
  02 num-requested TYPE BINARY 16 UNSIGNED VALUE 1.
  02 max-datasize TYPE BINARY 16 UNSIGNED VALUE 80.
  02 pad-char PIC X VALUE SPACE.
  02 filler PIC X VALUE SPACE.
END.

DEF read-prof-rec-a02-rsp.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 231.
  02 retn-code TYPE BINARY 16.
  02 retn-code-detail TYPE BINARY 16.
  02 corr-name PIC X(80).
  02 num-returned TYPE BINARY 16 UNSIGNED.
    OCCURS 0 TO !num-requested! 1 TIMES DEPENDING ON num-returned.
  02 recs-returned PIC X(max-datasize)! X(80).
  03 rec-type PIC 9(4) COMP.
  03 rec-seq-num PIC S9(4) COMP.
  03 update-control TYPE BINARY 16.
  03 data-len TYPE BINARY 16.
  03 data-string PIC !X(max-datasize)! X(80).
END.
```
READ-PROF-REC-A02 FIELDS. The fields defined in this UOW are:

- **HDR** is the UOW header. The UOW-CODE value is 231.

- **CORR-NAME** is the name of the correspondent whose depot profile record is to be read. This is also the name of the correspondent represented by the calling process. Only system administrators can enter correspondent names other than their own in this field; correspondents without system administrator privileges must enter their own names or leave the field blank. On input, you can use a partially qualified name, including wildcard characters. In the response, the fully qualified name is returned.

- **REC-TYPE** is the type assigned to the record by your application. In the response, the type of the record accessed for the retrieval is returned.

- **REC-SEQ-NUM** is the sequence number assigned to the starting record. In the response, the sequence number of the record retrieved is returned.

- **OPTIONS** provides criteria for selecting the profile records.
  
  **SKIP-EXACT** determines whether TISERV begins its retrieval with the record specified or with the record after the one specified.

  - **Y** = TISERV begins its retrieval with the record after the one specified; the record exactly matching the specified REC-TYPE and REC-SEQ-NUM is not returned.
  - **N** = TISERV begins its retrieval with the record specified.

  **ANY-REC-TYPE** determines whether records not having the record type specified in REC-TYPE also can be returned.

  - **Y** = All profile records can be returned; ANY-SEQ-NUM is ignored.
  - **N** = The result depends on the setting of ANY-SEQ-NUM.

  **ANY-SEQ-NUM** is meaningful only if **ANY-REC-TYPE** is set to **N**.

  - **ANY-REC-TYPE = N** and **ANY-SEQ-NUM = Y**: any subsequent profile records whose REC-TYPE field matches the given REC-TYPE field can be returned.
  - **ANY-REC-TYPE = N** and **ANY-SEQ-NUM = N**: at most one record whose REC-TYPE and REC-SEQ-NUM match the given values is returned.

  **DEPOT-FLAG** determines whether the UOW references a depot profile record or a system control record.
Y = a depot profile record, as indicated by CORR-NAME

N = a system control record

If you make no entry in this field, Y is assumed.

- NUM-REQUESTED specifies the number of records to be returned. This value directly affects the length of the response.

The records are returned starting with the lowest record type and lowest sequence number matching the OPTIONS selected. All records within a record type are returned in sequence number order before records of the next higher record type.

- MAX-DATASIZE refers to the data portion of each record; this value is the maximum number of bytes that can be retrieved. Records that are shorter than this length are automatically padded with the padding character specified by PAD-CHAR. Records that are longer than this length are truncated, but the length returned by DATA-LEN is the actual length prior to truncation. You should specify an even-numbered maximum length so that all elements of the returned array are word aligned; but if you specify an odd-numbered maximum length, TRANSFER automatically increments this value by 1.

MAX-DATASIZE, like NUM-REQUESTED, directly affects the length of the response.

The maximum value of MAX-DATASIZE is 1500 bytes.

- PAD-CHAR is the character used to pad returned records that are shorter than the maximum length specified by MAX-DATASIZE. The PAD-CHAR is usually a space.

- RETN-CODE is the return code. TISERV returns a code in this field to indicate one of the following entries.

  To indicate successful record retrieval:

  0 OK

  To indicate problems with the correspondent name:

  5600 E-CORR-NSRV-ERR
  5601 E-CORR-NOT-FOUND
  5602 E-CORR-BAD-NAME
  5604 E-CORR-NO-SUCH-NODE
  5606 E-CORR-NSRV-NOT-FOUND
  5607 E-CORR-NSRV-DOWN
  5611 E-CORR-NET-DOWN
  5622 E-CORR-NET-SAME-NODE
  5623 E-CORR-AMBIGUOUS-NAME
UOW Descriptions
READ-PROF-REC-A02

To indicate other problems:

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4001</td>
<td>W-EOF</td>
</tr>
<tr>
<td>4046</td>
<td>E-INVALID-REC-TYPE</td>
</tr>
<tr>
<td>4051</td>
<td>E-MUST-BE-YN</td>
</tr>
<tr>
<td>4086</td>
<td>W-DATASIZE-ADJUSTED</td>
</tr>
<tr>
<td>4087</td>
<td>E-INVALID-MAX-DATASIZE</td>
</tr>
<tr>
<td>4089</td>
<td></td>
</tr>
<tr>
<td>4092</td>
<td></td>
</tr>
<tr>
<td>4093</td>
<td></td>
</tr>
<tr>
<td>4201</td>
<td></td>
</tr>
</tbody>
</table>

- **RETN-CODE-DETAIL** is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

- **NUM-RETURNED** is the number of records returned by TISERV in the response.

- **RECS-RETURNED** is the data array that contains the records retrieved. For this array, you must define limits and element sizes that are consistent with the records requested in the UOW. Within this structure:

  - **REC-TYPE** contains the record type.
  - **REC-SEQ-NUM** contains the record sequence number.
  - **UPDATE-CONTROL** is the counter used to control concurrent updates of the profile record. If you are planning to update or delete a profile record, you must pass the **UPDATE-CONTROL** field returned by this UOW to the **WRITE-PROFILE-REC** or **DELETE-PROFILE-REC** UOW.
  - **DATA-LEN** contains the length of the data (non-key) portion of the record prior to retrieval, regardless of the value specified by **MAX-DATASIZE**.
  - **DATA-STRING** contains the data portion of the record. The length of the data in **DATA-STRING** can be modified, but should be consistent with the value in **MAX-DATASIZE**.

**READ-PROF-REC-A02 OPERATION.** READ-PROF-REC-A02 reads a user-maintained record from the depot Profile file and returns the data portion of the record. Although the profile record header is not returned, the record type and sequence number from this header are made available as separate fields (**REC-TYPE** and **REC-SEQ-NUM**).

Each record is padded or truncated to a fixed length before it is returned. The length of the **DATA-STRING** in the response is the value specified by **MAX-DATASIZE** in the request, rounded up to an even number.
The record is read from the depot (correspondent) identified by CORR-NAME, and is selected in accordance with REC-TYPE, REC-SEQ-NUM, and specified OPTIONS. If the DEPOT-FLAG was set to indicate a system control record, the single copy of this record in the system was read. The record is returned to the array named RECS-RETURNED.

The following warning conditions can occur:

**W-EOF**  The number of records satisfying the search criteria is less than or equal to NUM-REQUESTED. All records satisfying the search criteria are returned.

**W-DATA-TRUNCATED**  At least one record returned was longer than MAX-DATASIZE. A truncated version of that record was returned.

**W-DATASIZE-ADJUSTED**  An odd number was specified for MAX-DATASIZE. TRANSFER used a value that is one more than that specified.

If more than one warning condition occurs, the warning message that appears first in the preceding list is returned. If W-DATA-TRUNCATED and W-DATASIZE-ADJUSTED both occur, the message W-DATA-TRUNCATED is returned.
SAVE-ITEM saves an item in a folder that has any ordering discipline except APPLIC-DEFINED; the SAVE-ITEM-BY-KEY UOW must be used to save items in folders with APPLIC-DEFINED ordering.

SAVE-ITEM can still be used to save items in folders created by TRANSFER A01/A02. SAVE-ITEM differs from the SAVE-ITEM-B00 UOW in that it does not have an unsave option to have items automatically removed at a specified time; with SAVE-ITEM, you must use the UNSAVE-ITEM UOW to remove items from a folder.

SAVE-ITEM FIELDS. The fields defined in this UOW are:

- HDR is the UOW header. The UOW-CODE value is 110.
- ITEM-ID identifies the item to be saved.
- FOLDER-NAME is the name of the folder in which the item is to be saved. Special folders INBOX and WASTEBASKET can be entered in this field.
SAVE-ITEM

You can save items in the special folder named WASTEBASKET; TRANSFER automatically removes the items from this folder when the correspondent's session terminates. You can save items in the INBOX folder, but this is not recommended because TRANSFER already saves items in this folder when appropriate; clients can automatically remove packages from the INBOX folder when saving them elsewhere by including an UNSAVE-ITEM UOW in the interprocess message.

• RETN-CODE is the return code. TISERV returns a code in this field to indicate one of the following entries.

To indicate successful saving of the item:

0 OK

To indicate problems with the folder name:

5675 E-FLD-NSRV-ERR 5681 E-FLD-NSRV-NOT-FOUND
5676 E-FLD-NOT-FOUND 5682 E-FLD-NSRV-DOWN
5677 E-FLD-BAD-NAME 5686 E-FLD-NET-DOWN
5678 E-FLD-BAD-TYPE 5697 E-FLD-NOT-SAME-NODE
5679 E-FLD-NO-SUCH-NODE 5698 E-FLD-AMBIGUOUS-NAME

To indicate other problems:

4010 E-BAD-TRANSACTION 4105 E-CONCURRNT-FLD-UPDATE
4035 E-ITEM-NOT-FOUND 4106 E-BAD-ORD-CRITERIA
4049 W-REC-ALREADY-EXISTS 4906 E-ERR-ITEMDESC-FILE
4103 E-INCORRECT-FLD-ORDER 4912 E-ERR-FOLDER-FILE
4104 E-DUP-ORDERING-KEY 4922 E-ERR-INV-FOLDER-FILE

• RETN-CODE-DETAIL is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

SAVE-ITEM OPERATION. SAVE-ITEM saves the item identified by ITEM-ID in the folder identified by FOLDER-NAME at the correspondent depot. The saved item remains in the folder until you explicitly unsave it with the UNSAVE-ITEM UOW in this or a later session.

Items are stored according to the folder's ordering criteria. Saving operations are as follows.
UOW Descriptions
SAVE-ITEM

Folders ordered by Saving Operations

<table>
<thead>
<tr>
<th>Folders ordered by</th>
<th>Saving Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIME-SAVED (T)</td>
<td>The item is saved in chronological order.</td>
</tr>
<tr>
<td>CREATOR-NAME (C)</td>
<td>The item is saved in alphabetic order by creator name.</td>
</tr>
<tr>
<td>EARLIEST-DELIV-DATE (E)</td>
<td>A package header item that is unalterable is saved in earliest delivery date order. An item that is not a package header or is a package header and is alterable is saved in creation date order.</td>
</tr>
</tbody>
</table>

The following rules apply to SAVE-ITEM:

1. If a folder has an APPLIC-DEFINED ordering discipline, items must be saved with the SAVE-ITEM-BY-KEY UOW. If a SAVE-ITEM is issued on a folder with APPLIC-DEFINED ordering, the error E-INCORRECT-FLD-ORDER is returned.

2. If a folder was created by TRANSFER A01/A02, items are ordered in ascending sequence by TIME-SAVED (T), unless the folder ordering criteria was subsequently altered by the ALTER-FOLDER-ORDER UOW.

3. If a folder does not allow duplicate ordering keys, any attempt to save more than one item with the same ordering key returns the error E-DUP-ORDERING-KEY.

4. If a SAVE-ITEM UOW is issued concurrently with the ALTER-FOLDER-ORDER UOW, the error E-CONCURRNT-FLD-UPDATE is returned. This prevents an item from being saved with an incorrect ordering key.

5. If a SAVE-ITEM UOW is issued concurrently with the DELETE-FOLDER UOW, the error E-CONCURRNT-FLD-UPDATE is returned. This prevents an application from saving an item in a folder that is concurrently being deleted.
SAVE-ITEM-B00 (UOW Code 141)

SAVE-ITEM-B00 saves an item in a folder that has any ordering discipline except APPLIC-DEFINED; the SAVE-ITEM-BY-KEY UOW must be used to save items in folders with APPLIC-DEFINED ordering. SAVE-ITEM-B00 includes an unsave option to have items automatically removed from the folder at a specified time.

```
DEF save-item-b00-uow.
  02 hdr.
    03 self-ident  PIC AA VALUE "UW".
    03 uow-code    TYPE BINARY 16 UNSIGNED VALUE 141.
  02 item-id.
    03 dummy      PIC X(12).
  02 folder-name PIC X(80).
  02 flags.
    03 date-is-rel TYPE BOOLEAN.
    03 reserved-1 TYPE BOOLEAN VALUE "N".
    03 reserved-2 TYPE BOOLEAN VALUE "N".
    03 reserved-3 TYPE BOOLEAN VALUE "N".
    03 reserved-4 TYPE BOOLEAN VALUE "N".
    03 reserved-5 TYPE BOOLEAN VALUE "N".
    03 reserved-6 TYPE BOOLEAN VALUE "N".
    03 reserved-7 TYPE BOOLEAN VALUE "N".
  02 unsave-time.
    03 date-time   REDEFINES DATE-TIME.
    04 year       PIC 9(4).
    04 month      PIC 9(2).
    04 day-of-month PIC 9(2).
    04 hour       PIC 9(2).
    04 minute     PIC 9(2).
    04 second     PIC 9(2).
    03 delta-time PIC 9(4) COMP.
    04 quantity   PIC A.
    04 units      PIC X.
END.
```

```
DEF save-item-b00-rsp.
  02 hdr.
    03 self-ident  PIC AA VALUE "UW".
    03 uow-code    TYPE BINARY 16 UNSIGNED VALUE 141.
  02 retn-code   TYPE BINARY 16.
  02 retn-code-detail TYPE BINARY 16.
END.
```
UOW Descriptions
SAVE-ITEM-B00

SAVE-ITEM-B00 FIELDS. The fields defined in this UOW are:

- HDR is the UOW header. The UOW-CODE value is 141.
- ITEM-ID identifies the item to be saved.
- FOLDER-NAME is the name of the folder, with an ordering discipline other than APPLIC-DEFINED, in which the item is to be saved. Special folders INBOX and WASTEBASKET can be entered in this field.

You can save items in the special folder named WASTEBASKET; TRANSFER automatically removes the items from this folder when the session terminates. You can save items in the INBOX folder, but this is not recommended because TRANSFER already saves items in this folder when appropriate; clients can automatically remove packages from the INBOX folder when saving them elsewhere by including an UNSAVE-ITEM UOW in the interprocess message.

- FLAGS describes miscellaneous flags that are used by the SAVE-ITEM-B00 UOW.
  - DATE-IS-REL is a flag that indicates whether the given date is relative or absolute.
    
    Y = relative
    
    N = absolute
  
  RESERVED-1 through RESERVED-7 are reserved for use by Tandem; these fields must be set to N.

- UNSAVE-TIME is the time when the item will be automatically removed from the folder. This date can be relative or absolute, depending on the DATE-IS-REL flag.

  In DELTA-TIME, UNITS is the unit of time denoted in QUANTITY. UNITS can be set to D (for days), H (for hours), and M (for minutes). The date must be sometime in the future. If the date is in the past, the error E-PAST-DATE-TIME will be returned.

  To save the item in the folder and not have the item automatically unsaved, set

  DATE-IS-REL = Y
  UNITS = 0

- RETN-CODE is the return code. TISERV returns a code in this field to indicate one of the following entries.
To indicate successful saving of the item:

0 OK

To indicate problems with the folder name:

5675 E-FLD-NSRV-ERR
5676 E-FLD-NOT-FOUND
5677 E-FLD-BAD-NAME
5678 E-FLD-BAD-TYPE
5679 E-FLD-NO-SUCH-NODE
5681 E-FLD-NSRV-NOT-FOUND
5682 E-FLD-NSRV-DOWN
5686 E-FLD-NET-DOWN
5697 E-FLD-NOT-SAME-NODE
5698 E-FLD-AMBIGUOUS-NAME

To indicate other problems:

4010 E-BAD-TRANSACTION
4035 E-ITEM-NOT-FOUND
4045 E-TSCHED-UNAVAIL
4049 W-REC-ALREADY-EXIST
4051 E-MUST-BE-YN
4071 E-PAST-DATE-TIME
4072 E-UNITS-MUST-BE-DHM
4073 E-INVALID-DATE-TIME
4074 E-INVALID-REL-TIME-QTY
4103 E-INCORRECT-FLD-ORDER
4104 E-DUP-ORDERING-KEY
4105 E-CONCURRNT-FLD-UPDATE
4106 E-BAD-ORD-CRITERIA
4906 E-ERR-ITEM-DESC-FILE
4912 E-ERR-FOLDER-FILE
4922 E-ERR-INV-FOLDER-FILE

• RETN-CODE-DETAIL is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

SAVE-ITEM-B00 OPERATION. SAVE-ITEM-B00 saves the item identified by ITEM-ID in the folder identified by FOLDER-NAME at the correspondent depot.

The following rules apply to SAVE-ITEM-B00:

1. The folder in which an item is being saved must not have an ordering discipline of APPLIC-DEFINED. If a SAVE-ITEM-B00 is issued on a folder that has APPLIC-DEFINED ordering discipline, the error E-INCORRECT-FLD-ORDER is returned.

2. An item cannot be saved in the same folder more than once. If an application tries to save the same item in a folder where the item has already been saved, the warning W-REC-ALREADY-EXIST is returned; if the item was previously saved, the unsave time specified in a second request or in additional requests is disregarded.
3. To change an unsave time once the time has been set in a folder, you must:
   a. save the item in another folder; this is to prevent the item from being deleted immediately after the unsave, but before the resave
   b. resave the item in the desired folder with the new time
   c. unsave the item from the first folder, if desired.

4. If the folder does not allow duplicate ordering keys and an application attempts to save more than one item with the same ordering key, the error E-DUP-ORDERING-KEY is returned.

5. If the RETN-CODE is E-TSCHED-UNAVAIL, the item is not saved in the folder. The unsave could not be scheduled, so no saving was performed. To save the item, you must either wait for TSCHED to become available or save the item without an unsave date and time specified.

6. If a SAVE-ITEM-B00 UOW is issued concurrently with the ALTER-FOLDER-ORDER UOW, the error E-CONCURRNT-FLD-UPDATE is returned. This prevents an item from being saved with an incorrect ordering key.

7. If a SAVE-ITEM-B00 UOW is issued concurrently with the DELETE-FOLDER UOW, the error E-CONCURRNT-FLD-UPDATE is returned. This prevents an application from saving an item in a folder that is currently being deleted.
SAVE-ITEM-BY-KEY (UOW Code 139)

SAVE-ITEM-BY-KEY saves an item in a folder that has an ordering discipline of APPLIC-DEFINED. The ordering key used to save the item is supplied by the application.

```plaintext
DEF save-item-by-key-uow.
  02 hdr.
      03 self-ident          PIC AA VALUE "UW".
      03 uow-code            TYPE BINARY 16 UNSIGNED VALUE 139.
  02 item-id.
      03 dummy              PIC X(12).
  02 folder-name          PIC X(80).
  02 flags.
      03 date-is-rel        TYPE BOOLEAN.
      03 reserved-1         TYPE BOOLEAN VALUE "N".
      03 reserved-2         TYPE BOOLEAN VALUE "N".
      03 reserved-3         TYPE BOOLEAN VALUE "N".
      03 reserved-4         TYPE BOOLEAN VALUE "N".
      03 reserved-5         TYPE BOOLEAN VALUE "N".
      03 reserved-6         TYPE BOOLEAN VALUE "N".
      03 reserved-7         TYPE BOOLEAN VALUE "N".
  02 unsave-time.
      03 date-time.
          04 year            PIC 9(4).
          04 month           PIC 9(2).
          04 day-of-month     PIC 9(2).
          04 hour            PIC 9(2).
          04 minute          PIC 9(2).
          04 second          PIC 9(2).
      03 delta-time         REDEFINES DATE-TIME.
          04 quantity        PIC 9(4) COMP.
          04 units           PIC A.
          04 filler          PIC X.
  02 applic-key-len      TYPE BINARY 16 UNSIGNED.
  02 applic-key.
      03 element          PIC X
          OCCURS 0 TO 200 TIMES DEPENDING ON applic-key-len.
END.
```
SAVE-ITEM-BY-KEY FIELDS. The fields defined in this UOW are:

- **HDR** is the UOW header. The UOW-CODE value is 139.
- **ITEM-ID** identifies the item to be saved.
- **FOLDER-NAME** is the name of the folder, with an ordering discipline of APPLIC-DEFINED, in which the item is to be saved. Special folders INBOX and WASTEBASKET cannot be entered in this field.
- **FLAGS** describes miscellaneous flags that are used by the SAVE-ITEM-BY-KEY UOW.
  - **DATE-IS-REL** is a flag that indicates whether the given date is relative or absolute.
    - Y = relative
    - N = absolute
  - **RESERVED-1** through **RESERVED-7** are reserved for use by Tandem; these fields must be set to N.
- **UNSAFE-TIME** is the time when the item will be automatically removed from the folder. This date can be relative or absolute, depending on the DATE-IS-REL flag.
In DELTA-TIME, UNITS is the unit of time denoted in QUANTITY. UNITS can be set to D (for days), H (for hours), and M (for minutes). The date must be sometime in the future. If the date is in the past, the error E-PAST-DATE-TIME will be returned.

To save the item in the folder and not have the item automatically unsaved, set

\[ \text{DATE-IS-REL} = Y \]
\[ \text{UNITS} = 0 \]

- APPLIC-KEY-LEN is the length, in bytes, of the ordering key by which items are stored in the folder. The length must be a constant from 0 through 200 and must be consistent with the APPLIC-KEY field.

If variable APPLIC-KEY-LENs are used to save items within the same folder, ordering anomalies can exist in folders that are ordered in descending sequence or that allow duplicates; therefore, APPLIC-KEY-LEN should be fixed length for a given folder. The application should pad short keys with an application specific pad character.

If APPLIC-KEY-LEN is \(< 0\) or \(> 200\), the error E-INVALID-KEY-LEN is returned.

- APPLIC-KEY is the ordering key of the item. This key is supplied by the application. The length of this data must be consistent with APPLIC-KEY-LEN.

NOTE

UOWs must start on word boundaries. If APPLIC-KEY-LEN contains an odd value and other UOWs follow this one in the request, you must append a one-byte FILLER to APPLIC-KEY.

- RETN-CODE is the return code. TISERV returns a code in this field to indicate one of the following entries.

To indicate successful saving of the item:

\[ 0 \text{ OK} \]
UOW Descriptions
SAVE-ITEM-BY-KEY

To indicate problems with the folder:

5675 E-FLD-NSRV-ERR          5681 E-FLD-NSRV-NOT-FOUND
5676 E-FLD-NOT-FOUND          5682 E-FLD-NSRV-DOWN
5677 E-FLD-BAD-NAME           5686 E-FLD-NET-DOWN
5678 E-FLD-BAD-TYPE           5697 E-FLD-NOT-SAME-NODE
5679 E-FLD-NO-SUCH-NODE       5698 E-FLD-AMBIGUOUS-NAME
5680 E-FLD-SECURITY

To indicate other problems:

4010 E-BAD-TRANSACTION        4104 E-DUP-ORDERING-KEY
4035 E-ITEM-NOT-FOUND         4105 E-CONCURRT-FLD-UPDATE
4045 E-TSCHED-UNAVAIL         4106 E-BAD-ORD-CRITERIA
4049 W-REC-ALREADY-EXISTS     4906 E-ERR-ITEMDESC-FILE
4071 E-PAST-DATE-TIME         4912 E-ERR-FOLDER-FILE
4102 E-INVALID-KEY-LEN        4922 E-ERR-INV-FOLDER-FILE
4103 E-INCORRECT-FLD-ORDER

• RETN-CODE-DETAIL is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

SAVE-ITEM-BY-KEY OPERATION. SAVE-ITEM-BY-KEY saves the item identified by ITEM-ID in the folder identified by FOLDER-NAME at the correspondent depot.

The following rules apply to SAVE-ITEM-BY-KEY:

1. The folder in which the item is being saved must have an ordering discipline of APPLIC-DEFINED. If a SAVE-ITEM-BY-KEY UOW is issued on a folder with an ordering discipline other than APPLIC-DEFINED, the error E-INCORRECT-FLD-ORDER is returned.

2. An item cannot be saved in the same folder more than once. If an application tries to save the same item in a folder where the item has already been saved, the warning W-REC-ALREADY-EXISTS is returned. This applies even if APPLIC-KEY has a different value.

If the item was previously saved, the unsave time specified in a second request or in additional requests is disregarded.
3. If the folder does not allow duplicate ordering keys and an application attempts to save more than one item with the same ordering key, the error E-DUP-ORDERING-KEY is returned. For two ordering keys to be duplicates, they must be exactly the same length (APPLIC-KEY-LEN) and each character must match exactly.

4. If a SAVE-ITEM-BY-KEY UOW is issued concurrently with the ALTER-FOLDER-ORDER UOW, the error E-CONCURRNT-FLD-UPDATE is returned. This prevents an item from being saved with an incorrect ordering key.

5. If a SAVE-ITEM-BY-KEY UOW is issued concurrently with the DELETE-FOLDER UOW, the error E-CONCURRNT-FLD-UPDATE is returned. This prevents an application from saving an item in a folder that is currently being deleted.

6. If the RETN-CODE is E-TSCHED-UNAVAIL, the item is not saved in the folder. The unsave could not be scheduled, so no saving was performed. To save the item, you must either wait for TSCHED to become available or save the item without an unsave date and time specified.

7. To change an unsave time once the time has been set in a folder, you must:
   a. save the item in another folder; this is to prevent the item from being deleted immediately after the unsave, but before the resave
   b. resave the item in the desired folder with the new time
   c. unsave the item from the first folder, if desired.
UOW Descriptions
SCAN-FOLDER

SCAN-FOLDER (UOW Code 120)

SCAN-FOLDER scans the contents of a folder and retrieves the IDs of items in the folder. SCAN-FOLDER returns item IDs according to the ordering key by which items were saved in the folder. If an application scans a folder that is in APPLIC-DEFINED order and requires the ordering key of each item, the SCAN-FOLDER-BY-KEY UOW must be used.

SCAN-FOLDER differs from the SCAN-FOLDER-B00 UOW in that it does not return the unsave time of the item; you must use the SCAN-FOLDER-B00 UOW to have the unsave time returned.

```
DEF scan-folder-uow.
  02 hdr.
      03 self-ident PIC AA VALUE "UW".
      03 uow-code TYPE BINARY 16 UNSIGNED VALUE 120.
  02 folder-name PIC X(BO).
  02 item-id PIC X(12).
  02 dummy
  03 options.
      03 filter-by-item-type TYPE BOOLEAN.
      03 reserved-1 TYPE BOOLEAN VALUE "N".
      03 item-type TYPE BINARY 16 UNSIGNED VALUE 20.
  02 num-requested

END.
```

```
DEF scan-folder-rsp.
  02 hdr.
      03 self-ident PIC AA VALUE "UW".
      03 uow-code TYPE BINARY 16 UNSIGNED VALUE 120.
  02 retn-code TYPE BINARY 16.
  02 retn-code-detail TYPE BINARY 16.
  02 num-returned TYPE BINARY 16 UNSIGNED.
  02 items-returned.
      03 item-id.
          04 dummy OCCURS 0 TO !num-requested! 20 TIMES DEPENDING ON num-returned.

END.
```
SCAN-FOLDER FIELDS. The fields defined in this UOW are:

- HDR is the UOW header. The UOW-CODE value is 120.

- FOLDER-NAME is the name of the folder whose contents are to be scanned.

- ITEM-ID identifies the item in the folder following which the retrieval is to start. You can begin with the first item in the folder by setting this field to binary zeros. If the starting ITEM-ID is nonzero, it must be the item ID of an item that is stored in the folder.

- OPTIONS allows you to filter the scan by selecting only items of a given type from the folder, as follows:
  1. To select this filtering, set FILTER-BY-ITEM-TYPE to Y. To suppress filtering, set this field to N.
  2. If you have selected filtering, specify the item type to be retrieved in ITEM-TYPE.

  The RESERVED-1 field in the OPTIONS structure is reserved for use by Tandem; this field must be set to N.

- NUM-REQUESTED is the number of items to be retrieved from the folder. The suggested maximum is 200. Based on a maximum IPC size of 3000 bytes containing the reply for a single SCAN-FOLDER UOW, about 200 items can be contained in the SCAN-FOLDER response; this estimate includes IPC and UOW response overhead with the result rounded down. The maximum size for an IPC is application dependent, although 3000 bytes is a reasonable upper bound.

- RETN-CODE is the return code. TISERV returns a code in this field to indicate one of the following entries.

  To indicate successful scanning of the folder:

    0 OK

  To indicate problems with the folder name:

    5675 E-FLD-NSRV-ERR
    5676 E-FLD-NOT-FOUND
    5677 E-FLD-BAD-NAME
    5678 E-FLD-BAD-TYPE
    5679 E-FLD-NO-SUCH-NODE
    5680 E-FLD-SECURITY
    5681 E-FLD-NSRV-NOT-FOUND
    5682 E-FLD-NSRV-DOWN
    5683 E-FLD-NO-PARENT
    5686 E-FLD-AMBIGUOUS-NAME
    5697 E-FLD-AMBIGUOUS-NAME
UOW Descriptions
SCAN-FOLDER

To indicate other errors:

<table>
<thead>
<tr>
<th>-ret</th>
<th>desc</th>
<th>ret</th>
<th>desc</th>
</tr>
</thead>
<tbody>
<tr>
<td>4001</td>
<td>W-EOF</td>
<td>4092</td>
<td>E-INVALID-NUM-RQSTD</td>
</tr>
<tr>
<td>4047</td>
<td>E-REC-NOT-FOUND</td>
<td>4105</td>
<td>E-CONCURRNT-FLD-UPDATE</td>
</tr>
<tr>
<td>4051</td>
<td>E-MUST-BE-YN</td>
<td>4912</td>
<td>E-ERR-FOLDER-FILE</td>
</tr>
<tr>
<td>4052</td>
<td>E-RESERVED-MUST-BE-N</td>
<td>4922</td>
<td>E-ERR-INV-FOLDER-FILE</td>
</tr>
<tr>
<td>4056</td>
<td>E-INVALID-ITEM-TYPE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- RETN-CODE-DETAIL is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

- NUM-RETURNED is the number of items retrieved from the folder by TRANSFER.

- ITEMS-RETURNED is an array that contains the IDs of the items retrieved from the folder. The size of this array must be consistent with the number of items requested, as defined in NUM-REQUESTED.

SCAN-FOLDER OPERATION. SCAN-FOLDER retrieves the IDs of the items in the range denoted by ITEM-ID and NUM-REQUESTED and places these IDs in the array named ITEMS-RETURNED. Items are returned in the order that they are saved in a folder.

The OPTIONS structure allows you to selectively retrieve only items of a particular type.

The number of items returned might be smaller than the number requested because the folder contains fewer items than specified in NUM-REQUESTED; in this case, the W-EOF warning is returned in the RETN-CODE field.

You can use the last item ID returned in ITEMS-RETURNED as input in ITEM-ID for the next request.
SCAN-FOLDER-B00 (UOW Code 142)

SCAN-FOLDER-B00 scans the contents of a folder and retrieves the IDs and unsave time of items in the folder. SCAN-FOLDER-B00 returns item IDs according to the ordering key by which items were saved in the folder. If an application scans a folder that is in APPLIC-DEFINED order and requires the ordering key of each item, the SCAN-FOLDER-BY-KEY UOW must be used.

```
DEF scan-folder-b00-uow.
  02 hdr.
     03 self-ident PIC AA VALUE "UW".
     03 uow-code TYPE BINARY 16 UNSIGNED VALUE 142.
  02 folder-name PIC X(80).
  02 item-id.
     03 dummy PIC X(12).
  02 options.
     03 filter-by-item-type TYPE BOOLEAN.
     03 reserved-1 TYPE BOOLEAN VALUE "N".
     03 item-type TYPE BINARY 16 UNSIGNED VALUE 20.
  02 num-requested
END.

DEF scan-folder-b00-rsp.
  02 hdr.
     03 self-ident PIC AA VALUE "UW".
     03 uow-code TYPE BINARY 16 UNSIGNED VALUE 142.
  02 retn-code TYPE BINARY 16.
  02 retn-code-detail TYPE BINARY 16.
  02 num-returned TYPE BINARY 16 UNSIGNED.
  02 items-returned OCCURS 0 TO !num-requested! TIMES DEPENDING ON num-requested.
  03 item-id.
     04 dummy PIC X(12).
  03 unsave-time.
     04 date-time
        05 year PIC 9(4).
        05 month PIC 9(2).
        05 day-of-month PIC 9(2).
        05 hour PIC 9(2).
        05 minute PIC 9(2).
        05 second PIC 9(2).
END.
```
SCAN-FOLDER-B00 FIELDS. The fields defined in this UOW are:

- **HDR** is the UOW header. The UOW-CODE value is 142.
- **FOLDER-NAME** is the name of the folder whose contents are to be scanned.
- **ITEM-ID** identifies the item in the folder following which the retrieval is to start. You can begin with the first item in the folder by setting this field to binary zeros. If the starting ITEM-ID is nonzero, it must be the item ID of an item that is stored in the folder.
- **OPTIONS** allows you to filter the scan by selecting only items of a given type from the folder, as follows:
  1. To select this filtering, set FILTER-BY-ITEM-TYPE to Y. To suppress filtering, set this field to N.
  2. If you have selected filtering, specify the item type to be retrieved in ITEM-TYPE.

The RESERVED-1 field in the OPTIONS structure is reserved for use by Tandem; this field must be set to N.

- **NUM-REQUESTED** is the number of items to be retrieved from the folder. The suggested maximum is 100. Based on a maximum IPC size of 3000 bytes containing the reply for a single SCAN-FOLDER-B00 UOW, about 100 items can be contained in the SCAN-FOLDER-B00 response; this estimate includes IPC and UOW response overhead with the result rounded down. The maximum size for an IPC is application dependent, although 3000 bytes is a reasonable upper bound.

- **RETN-CODE** is the return code. TISERV returns a code in this field to indicate one of the following entries.

  To indicate successful scanning of the folder:

  0 OK

  To indicate problems with the folder name:

  5675 E-FLD-NSRV-ERR
  5676 E-FLD-NOT-FOUND
  5677 E-FLD-BAD-NAME
  5678 E-FLD-BAD-TYPE
  5679 E-FLD-NO-SUCH-NODE
  5680 E-FLD-SECURITY
  5681 E-FLD-NSRV-NOT-FOUND
  5682 E-FLD-NSRV-DOWN
  5683 E-FLD-NO-PARENT
  5686 E-FLD-NET-DOWN
  5697 E-FLD-NOT-SAME-NODE
  5698 E-FLD-AMBIGUOUS-NAME
To indicate other errors:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4001</td>
<td>W-EOF</td>
<td>4092</td>
<td>E-INVALID-NUM-RQSTD</td>
</tr>
<tr>
<td>4047</td>
<td>E-REC-NOT-FOUND</td>
<td>4105</td>
<td>E-CONCURRNT-FLD-UPDATE</td>
</tr>
<tr>
<td>4051</td>
<td>E-MUST-BE-YN</td>
<td>4912</td>
<td>E-ERR-FOLDER-FILE</td>
</tr>
<tr>
<td>4052</td>
<td>E-RESERVED-MUST-BE-N</td>
<td>4922</td>
<td>E-ERR-INV-FOLDER-FILE</td>
</tr>
<tr>
<td>4056</td>
<td>E-INVALID-ITEM-TYPE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- RETN-CODE-DETAIL is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.
- NUM-RETURNED is the number of items retrieved from the folder by TRANSFER.
- ITEMS-RETURNED is an array that contains the IDs and the unsave time of the items retrieved from the folder. The size of this array must be consistent with the number of items requested, as defined in NUM-REQUESTED.
- UNSAVE-TIME is the time when the item will be automatically removed from the folder. The date and time returned is in absolute format. If the item has no unsave date and time, all fields of UNSAVE-TIME are set to 0.

SCAN-FOLDER-B00 OPERATION. SCAN-FOLDER-B00 retrieves the IDs and unsave times of the items in the range denoted by ITEM-ID and NUM-REQUESTED and places these values in the array named ITEMS-RETURNED. Items are returned in ordering key sequence.

The OPTIONS structure allows you to selectively retrieve only items of a particular type.

The number of items returned might be smaller than the number requested because the folder contains fewer items than specified in NUM-REQUESTED; in this case, the W-EOF warning is returned in the RETN-CODE field.

You can use the last item ID returned in ITEMS-RETURNED as input in ITEM-ID for the next request.
SCAN-FOLDER-BY-KEY retrieves the item ID, item type, unsave time, and the corresponding ordering key and key length of items saved in folders with APPLIC-DEFINED ordering. Items are returned in application ordering key sequence.

```
DEF scan-folder-by-key-uow.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 140.
    02 folder-name PIC X(80).
  02 options.
    03 filter-by-item-type TYPE BOOLEAN.
    03 skip-exact TYPE BOOLEAN.
    03 duplicate-restart TYPE BOOLEAN.
    03 reserved-3 TYPE BOOLEAN VALUE "N".
    03 reserved-4 TYPE BOOLEAN VALUE "N".
    03 reserved-5 TYPE BOOLEAN VALUE "N".
    03 reserved-6 TYPE BOOLEAN VALUE "N".
    03 reserved-7 TYPE BOOLEAN VALUE "N".
    03 item-type PIC 9(4) COMP.
    02 num-requested TYPE BINARY 16 UNSIGNED VALUE 10.
    02 starting-key-len TYPE BINARY 16 UNSIGNED.
    02 generic-key-len TYPE BINARY 16 UNSIGNED.
    02 max-key-len TYPE BINARY 16.
    02 pad-char PIC X VALUE SPACES.
    02 starting-duplicate PIC X(10) VALUE LOW-VALUES.
    02 starting-key PIC X(200).
    02 filler TYPE BINARY 16 UNSIGNED VALUE 0.
END.
```

```
DEF scan-folder-by-key-rsp.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 140.
    02 retn-code TYPE BINARY 16.
    02 retn-code-detail TYPE BINARY 16.
    02 folder-name PIC X(80).
END.
```
UOW Descriptions
SCAN-FOLDER-BY-KEY

02 options.
  03 filter-by-item-type TYPE BOOLEAN.
  03 skip-exact TYPE BOOLEAN.
  03 duplicate-restart TYPE BOOLEAN.
  03 reserved-3 TYPE BOOLEAN VALUE "N".
  03 reserved-4 TYPE BOOLEAN VALUE "N".
  03 reserved-5 TYPE BOOLEAN VALUE "N".
  03 reserved-6 TYPE BOOLEAN VALUE "N".
  03 reserved-7 TYPE BOOLEAN VALUE "N".
  03 item-type PIC 9(4) COMP.

02 num-requested TYPE BINARY 16 UNSIGNED VALUE 10.
  02 starting-key-len TYPE BINARY 16 UNSIGNED.
  02 generic-key-len TYPE BINARY 16 UNSIGNED.
  02 max-key-len TYPE BINARY 16 UNSIGNED.
  02 pad-char PIC X VALUE SPACES.
  02 starting-duplicate PIC X(10) VALUE LOW-VALUES.
  02 starting-key PIC X(200).
  02 num-returned TYPE BINARY 16 UNSIGNED VALUE 0.

02 items-returned OCCURS 0 TO !num-requested! 10 TIMES DEPENDING ON num-returned.

03 item-id.
  04 dummy PIC X(12).

03 item-type PIC 9(4) COMP.

03 unsave-time.
  04 date-time.
    05 year PIC 9(4).
    05 month PIC 9(2).
    05 day-of-month PIC 9(2).
    05 hour PIC 9(2).
    05 minute PIC 9(2).
    05 second PIC 9(2).

03 ordering-key-length TYPE BINARY 16 UNSIGNED.

03 ordering-key PIC !X(MAX-KEY-LEN)! X(200).

END.

SCAN-FOLDER-BY-KEY FIELDS. The fields defined in this UOW are:

• HDR is the UOW header. The UOW-CODE value is 140.
UOW Descriptions
SCAN-FOLDER-BY-KEY

• FOLDER-NAME is the name of the folder whose contents are to be scanned.

• OPTIONS provides criteria for selecting items.

  FILTER-BY-ITEM-TYPE allows you to filter the scan by selecting only items of a given type from the folder.
  - To select filtering, set FILTER-BY-ITEM-TYPE to Y. To suppress filtering, set this field to N.
  - If you have selected filtering, specify the item type to be retrieved in ITEM-TYPE.

In the response, the input to FILTER-BY-ITEM-TYPE and ITEM-TYPE is echoed as output.

SKIP-EXACT indicates whether TISERV begins its retrieval with the item specified by the starting key fields or with the next item in the folder.

  Y = Start retrieval with the first item following the item specified by the starting key fields (if that item is present), or with the first item following (if the exact item is not present).

  N = Start retrieval with the exact item specified by the starting key fields (if that item is present), or with the first item following (if the exact item is not present).

In the response, SKIP-EXACT is set to Y.

DUPLICATE-RESTART controls the use of the STARTING-DUPLICATE field.

  Y = The STARTING-DUPLICATE field is used in conjunction with STARTING-KEY to identify the first item to return.

  N = The STARTING-DUPLICATE field is ignored.

DUPLICATE-RESTART contains restart information. On the initial call to SCAN-FOLDER-BY-KEY, DUPLICATE-RESTART is typically set to N. In the response, DUPLICATE-RESTART will be automatically set according to the last item returned in the current scan; this will be the appropriate setting for DUPLICATE-RESTART in the request for a subsequent scan of the same folder.
The RESERVED-3 through RESERVED-7 fields in the OPTIONS structure are reserved for use by Tandem; these fields are always set to N.

- NUM-REQUESTED is the number of items to be retrieved from the folder. The value of NUM-REQUESTED affects the size of the response UOW.

The maximum of this number is dependent on the value of MAX-KEY-LEN. The suggested maximum for a MAX-KEY-LEN of 200 would be 10. Based on a MAX-KEY-LEN of 200 and a maximum IPC size of 3000 bytes containing the reply for a single SCAN-FOLDER-BY-KEY UOW, about 10 items can be contained in the SCAN-FOLDER-BY-KEY response; this estimate includes IPC and UOW response overhead with the result rounded down. The maximum size for an IPC is application dependent, although 3000 bytes is a reasonable upper bound.

In the response, the input to NUM-REQUESTED is echoed as output.

- STARTING-KEY-LEN is the length, in bytes, of the initial ordering key information in the STARTING-KEY field. This field together with STARTING-KEY and STARTING-DUPLICATE determines which item is the first retrieved. The first item in the folder with an ordering key either equal to or greater than STARTING-KEY for STARTING-KEY-LEN bytes is the first item returned in the ITEMS-RETURNED field. Only the first STARTING-KEY-LEN bytes of STARTING-KEY are used; the remainder of the field is ignored.

To retrieve the first item in the folder, set this field to 0; when length is set to 0, GENERIC-KEY-LEN and SKIP-EXACT are ignored.

The maximum value for STARTING-KEY-LEN is 200. If this field is less than 0 or greater than 200, the error E-INVALID-KEY-LEN is returned.

In the response, STARTING-KEY-LEN will correspond to the last item returned in the current scan. This is the appropriate setting for the STARTING-KEY-LEN field in the request for a subsequent scan of the same folder.
UOW Descriptions
SCAN-FOLDER-BY-KEY

- GENERIC-KEY-LEN is the length, in bytes, of the STARTING-KEY to be used for generic matching. The maximum value for GENERIC-KEY-LEN is 200. If this field is less than zero or greater than 200, the error E-INVALID-KEY-LEN is returned.

  If GENERIC-KEY-LEN is > 0, only items with ordering keys matching STARTING-KEY for GENERIC-KEY-LEN bytes will be returned.

  If GENERIC-KEY-LEN = 0, generic matching is not used.

  If this field is not less than or equal to STARTING-KEY-LEN, the error E-INVALID-KEY-LEN is returned.

  In the response, input to this field is echoed as output.

- MAX-KEY-LEN refers to the ORDERING-KEY field returned with each item in ITEM-RETURNED. This is the maximum number of bytes in ORDERING-KEY to be returned. The maximum value for MAX-KEY-LEN is 200. If this field is less than zero or greater than 200, the error E-INVALID-KEY-LEN is returned.

  Ordering keys that are shorter than MAX-KEY-LEN are automatically padded with the padding character specified by the PAD-CHAR field.

  Ordering keys that are longer than MAX-KEY-LEN are truncated, and the warning W-ORD-KEY-TRUNCATED is returned. The length returned in the corresponding ORDERING-KEY-LEN field, however, is the actual length prior to truncation.

  MAX-KEY-LEN should be an even number length so that all elements of ITEMS-RETURNED are word aligned. If the number in MAX-KEY-LEN is odd, TISERV automatically increments this value by 1 and returns the warning W-MAX-KEY-LEN-ADJUSTED. This warning, however, is overridden by the warnings W-EOF and W-ORD-KEY-TRUNCATED.

  In the response, the input to this field is echoed as output.

- PAD-CHAR is the character to be used for padding the returned ORDERING-KEY when the ordering key is shorter than MAX-KEY-LEN. In the response, the input to this field is echoed as output.

- STARTING-DUPLICATE is used with folders that contain duplicate ordering keys. This field contains an internal restart value that is used to distinguish between items with duplicate ordering keys.

  STARTING-DUPLICATE is typically not used on the initial call to SCAN-FOLDER-BY-KEY (DUPLICATE-RESTART is typically set to N, which causes STARTING-DUPLICATE to be ignored).
If STARTING-DUPLICATE is used, STARTING-KEY must contain the full ordering key as returned in the response STARTING-KEY field. In the response, STARTING-DUPLICATE will correspond to the last item returned in the current scan. This is the appropriate setting for the STARTING-DUPLICATE field in the request for a subsequent scan of the same folder.

- STARTING-KEY is the ordering key of the first item to retrieve. The size must be consistent with STARTING-KEY-LEN. In the response, STARTING-KEY will correspond to the last item returned in the current scan. This is the appropriate setting for the STARTING-KEY field in the request for a subsequent scan of the same folder.

- RETN-CODE is the return code. TISERV returns a code in this field to indicate one of the following entries.

To indicate successful retrieval of items:

0 OK

To indicate problems with the folder name:

5675 E-FLD-NSRV-ERR
5676 E-FLD-NOT-FOUND
5677 E-FLD-BAD-NAME
5678 E-FLD-BAD-TYPE
5679 E-FLD-NO-SUCH-NODE
5680 E-FLD-SECURITY
5681 E-FLD-NSRV-NOT-FOUND
5682 E-FLD-NSRV-DOWN
5683 E-FLD-NO-PARENT
5686 E-FLD-NET-DOWN
5697 E-FLD-NOT-SAME-NODE
5698 E-FLD-AMBIGUOUS-NAME

To indicate other problems:

4001 W-EOF
4051 E-MUST-BE-YN
4052 E-RESERVED-MUST-BE-N
4056 E-INVALID-ITEM-TYPE
4092 E-INVALID-NUM-RQSTD
4100 W-ORD-KEY-TRUNCATED
4101 W-MAX-KEY-LEN-ADJUSTED
4102 E-INVALID-KEY-LEN
4103 E-INCORRECT-FLD-ORDER
4105 E-CONCURRNT-FLD-UPDATE
4106 E-BAD-ORD-CRITERIA
4912 E-ERR-FOLDER-FILE

- RETN-CODE-DETAIL is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

- NUM-RETURNED is the number of items retrieved from the folder. The number of items retrieved might be less than the number of items requested (NUM-REQUESTED) because:

  either the folder contains fewer items in the range specified by the passed starting key information or the folder contains fewer items in the subset specified for generic matching.
UOW Descriptions
SCAN-FOLDER-BY-KEY

If this occurs, the warning W-EOF is returned in the RETN-CODE field.

• ITEMS-RETURNED is an array containing the item ID (ITEM-ID), item type (ITEM-TYPE), ordering key length (ORDERING-KEY-LEN), and ordering key (ORDERING-KEY) of the items returned. The size of this array must be consistent with the number of items requested, as defined in NUM-REQUESTED.

• UNSAVE-TIME is the time when the item will be automatically removed from the folder. The date and time returned is in absolute format. If the item has no unsave date and time, all fields of UNSAVE-TIME are set to 0.

SCAN-FOLDER-BY-KEY OPERATION. SCAN-FOLDER-BY-KEY retrieves the item IDs, item types, unsave time, and the corresponding ordering key and key length of the items in the range denoted by STARTING-KEY, STARTING-DUPLICATE, and NUM-REQUESTED. These values are placed in the array named ITEMS-RETURNED. Items are returned in application ordering key sequence.

NOTE

This UOW is restricted to folders with APPLIC-DEFINED ordering disciplines. If this UOW is attempted with a folder that has an ordering discipline other than APPLIC-DEFINED, the error E-INCORRECT-FLD-ORDER is returned.

The OPTIONS structure allows you to selectively retrieve only items of a particular type. The structure also allows you to retrieve either the first matching item or the item immediately following.

The SCAN-FOLDER-BY-KEY UOW allows the application to use the response definition to build the request. VALUE clauses are appropriately set to allow this usage. NUM-RETURNED must be set to 0 when the request is issued; otherwise the request will be the wrong length. Note that the actual value of NUM-RETURNED does not matter; the length of the UOW sent, however, must be correct. In higher level languages, this implies that NUM-RETURNED must be set to 0.
Figure 5-7 illustrates the operation of SCAN-FOLDER-BY-KEY.

<table>
<thead>
<tr>
<th>Item</th>
<th>Ordering Key Length</th>
<th>Ordering Key</th>
<th>Duplicate Indicator</th>
</tr>
</thead>
<tbody>
<tr>
<td>first-item</td>
<td>4</td>
<td>TERM</td>
<td></td>
</tr>
<tr>
<td>second-item</td>
<td>4</td>
<td>TERM</td>
<td>first-dup</td>
</tr>
<tr>
<td>third-item</td>
<td>4</td>
<td>TERM</td>
<td>second-dup</td>
</tr>
<tr>
<td>fourth-item</td>
<td>7</td>
<td>TERM244</td>
<td></td>
</tr>
<tr>
<td>fifth-item</td>
<td>7</td>
<td>TERM245</td>
<td></td>
</tr>
<tr>
<td>sixth-item</td>
<td>7</td>
<td>TERM246</td>
<td></td>
</tr>
</tbody>
</table>

A SCAN-FOLDER-BY-KEY UOW is issued with values set in the following fields:

- **SKIP-EXACT** = N  suggested value for initial call
- **DUPLICATE-RESTART** = N  suggested value for initial call
- **NUM-REQUESTED** = 5  request for 5 items
- **STARTING-KEY-LEN** = 0  retrieve the first item in the folder
- **GENERIC-KEY-LEN** = 0  generic matching is not used
- **MAX-KEY-LEN** = 6  maximum number of bytes to be retrieved
- **PAD-CHAR** = *  ordering keys shorter than MAX-KEY-LEN to be padded with * character
- **STARTING-DUPLICATE** =  ignored; **DUPLICATE-RESTART** = N
- **STARTING-KEY** = ignored; **STARTING-KEY-LEN** = 0

Figure 5-7. SCAN-FOLDER-BY-KEY UOW Example
TISERV begins retrieval, encounters the first item (TERM), and sets the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SKIP-EXACT</td>
<td>Y</td>
<td>set for next scan</td>
</tr>
<tr>
<td>DUPLICATE-RESTART</td>
<td>N</td>
<td>first-item is not preceded by a duplicate</td>
</tr>
<tr>
<td>STARTING-DUPLICATE</td>
<td></td>
<td>ignored; DUPLICATE-RESTART = N</td>
</tr>
<tr>
<td>STARTING-KEY</td>
<td>TERM</td>
<td>ordering key value of first-item</td>
</tr>
<tr>
<td>STARTING-KEY-LEN</td>
<td>4</td>
<td>ordering key length of first-item</td>
</tr>
</tbody>
</table>

TISERV encounters the second item (TERM), and sets the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SKIP-EXACT</td>
<td>Y</td>
<td>set for next scan</td>
</tr>
<tr>
<td>DUPLICATE-RESTART</td>
<td>Y</td>
<td>second-item is a duplicate</td>
</tr>
<tr>
<td>STARTING-DUPLICATE</td>
<td>first-dup</td>
<td>internal duplicate indicator</td>
</tr>
<tr>
<td>STARTING-KEY</td>
<td>TERM</td>
<td>ordering key value of second-item</td>
</tr>
<tr>
<td>STARTING-KEY-LEN</td>
<td>4</td>
<td>ordering key length of second-item</td>
</tr>
</tbody>
</table>

TISERV encounters the third item (TERM), and sets the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SKIP-EXACT</td>
<td>Y</td>
<td>set for next scan</td>
</tr>
<tr>
<td>DUPLICATE-RESTART</td>
<td>Y</td>
<td>third-item is a duplicate</td>
</tr>
<tr>
<td>STARTING-DUPLICATE</td>
<td>second-dup</td>
<td>internal duplicate indicator</td>
</tr>
<tr>
<td>STARTING-KEY</td>
<td>TERM</td>
<td>ordering key value of third-item</td>
</tr>
<tr>
<td>STARTING-KEY-LEN</td>
<td>4</td>
<td>ordering key length of third-item</td>
</tr>
</tbody>
</table>

Figure 5-7. SCAN-FOLDER-BY-KEY UOW Example (Continued)
TISERV encounters the fourth item (TERM244), and sets the following fields:

- **SKIP-EXACT** = Y set for next scan
- **DUPLICATE-RESTART** = N fourth-item is not preceded by a duplicate
- **STARTING-DUPLICATE** = ignored; **DUPLICATE-RESTART** = N
- **STARTING-KEY** = TERM244 ordering key value of fourth-item
- **STARTING-KEY-LEN** = 7 ordering key length of fourth-item

TISERV encounters the fifth item (TERM245), and sets the following fields:

- **SKIP-EXACT** = Y set for next scan
- **DUPLICATE-RESTART** = N fifth-item is not preceded by a duplicate
- **STARTING-DUPLICATE** = ignored; **DUPLICATE-RESTART** = N
- **STARTING-KEY** = TERM245 ordering key value of fifth-item
- **STARTING-KEY-LEN** = 7 ordering key length of fifth-item

These returned values for the fifth item (TERM245) are the appropriate settings for the corresponding request fields for a subsequent scan of the same folder.

TISERV sets **NUM-RETURNED** to 5 and outputs the following entries in the **ITEMS-RETURNED** structure:

<table>
<thead>
<tr>
<th>Item</th>
<th>Ordering Key Length</th>
<th>Ordering Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>first-item</td>
<td>4</td>
<td>TERM**</td>
</tr>
<tr>
<td>second-item</td>
<td>4</td>
<td>TERM**</td>
</tr>
<tr>
<td>third-item</td>
<td>4</td>
<td>TERM**</td>
</tr>
<tr>
<td>fourth-item</td>
<td>7</td>
<td>TERM24</td>
</tr>
<tr>
<td>fifth-item</td>
<td>7</td>
<td>TERM24</td>
</tr>
</tbody>
</table>

Figure 5-7. SCAN-FOLDER-BY-KEY UOW Example (Continued)
START-SESSION

START-SESSION (UOW Code 101)

START-SESSION starts a session.

DEF start-session-uow.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 101.
  02 applic-id PIC 9(4) COMP.
  02 corr-name PIC X(120).
  02 password PIC X(16).
  02 curr-local-time.
    03 year PIC 9(4).
    03 month PIC 9(2).
    03 day-of-month PIC 9(2).
    03 hour PIC 9(2).
    03 minute PIC 9(2).
    03 second PIC 9(2).
  02 redef-local-time REDEFINES CURR-LOCAL-TIME.
    03 CENTURY PIC 9(2).
    03 ACCEPT-DATE PIC 9(6).
    03 ACCEPT-TIME PIC 9(6).
END.

DEF start-session-rsp.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 101.
  02 retn-code TYPE BINARY 16.
  02 retn-code-detail TYPE BINARY 16.
  02 filler PIC X.
  02 server-version.
    03 letter PIC A.
    03 rev-number PIC 99.
  02 time-zone-diff PIC S9(4) COMP.
  02 resolved-name PIC X(120).
END.

START-SESSION FIELDS. The fields defined in this UOW are:

- HDR is the UOW header. The UOW-CODE value is 101.
• APPLIC-ID is the numeric application ID, as defined by your application. All packages created during this session inherit this attribute as part of their item descriptor; this field can then be used as an agent selection criterion at depots receiving this package. In this field, the values 100 through 999 are reserved for use by Tandem.

• CORR-NAME is the name of the correspondent represented by the requesting process. This name must be previously defined in the system through the CREATE-DEPOT UOW.

• PASSWORD is the password required to establish the session, as defined in the system by the ALTER-PROFILE-ELEM UOW.

• CURR-LOCAL-TIME is the current date and time as obtained by the requesting process. TRANSFER uses this field to determine whether a time difference exists between requester and server. If a difference does exist, TRANSFER applies it to the adjustment of various timestamps used in the delivery of packages.

• REDEF-LOCAL-TIME redefines the CURR-LOCAL-TIME field for SCREEN COBOL coding convenience.

• RETN-CODE is the return code. TISERV returns a code in this field to indicate one of the following entries.

  To indicate successful starting of the session:

  0 OK

  To indicate problems with the correspondent name:

  5600 E-CORR-NSRV-ERR 5606 E-CORR-NSRV-NOT-FOUND
  5601 E-CORR-NOT-FOUND 5607 E-CORR-NSRV-DOWN
  5602 E-CORR-BAD-NAME 5611 E-CORR-NET-DOWN
  5603 E-CORR-BAD-TYPE 5622 E-CORR-NOT-SAME-NODE
  5604 E-CORR-NO-SUCH-NODE 5623 E-CORR-AMBIGUOUS-NAME
  5605 E-CORR-SECURITY 5624 E-CORR-BAD-SUFFIX

  To indicate other problems:

  4005 W-CONCURRENT-SESSION 4017 E-RESTRICTED-OPERATION
  4005 E-CONCURRENT-SESSION 4019 E-ALREADY-IN-SESSION
  4007 E-LOGON-DISALLOWED 4021 E-INVALID-ZONE-OFFSET
  4010 E-BAD-TRANSACTION 4055 E-INVALID-APPLIC-ID
  4013 E-NO-DEPOT-FOR-CORR 4073 E-INVALID-DATE-TIME
  4015 E-INVALID-PASSWORD 4201 E-CONTEXT-ERR

• RETN-CODE-DETAIL is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.
UOW Descriptions

START-SESSION

- **SERVER-VERSION** is the software version designation of TISERV, returned by TRANSFER. In the first release, this is A01.

- **TIME-ZONE-DIFF** is the difference between the current timestamps of the server and the requester. This value is expressed in minutes and rounded to the nearest minute.

Whenever a requester establishes a TRANSFER session, the difference in current times between server and requester is retained as one of the session attributes. TRANSFER uses this time zone difference to translate all timestamps into the local data base frame of reference prior to storage, and to translate them back to the local time of the requester prior to replying. Thus, requesters always communicate with TRANSFER in their own local timeframe, and all timestamps are stored relative to the data base local time.

The same transformations are applied to any stored time zone differences. The sender time zone difference stored in a package header is initially set to the session (server/requester) time zone difference when the package is submitted. Whenever the package is transferred from one node to another, the following adjustments are made at the receiving node: the difference in local times between the receiving node and the transferring node is added to all timestamps (except the delivered timestamp) and to the time zone difference field.

- **RESOLVED-NAME** is the resolved name of the correspondent, returned by TRANSFER. If this is a remote name, the session will not be established; however, the returned name can be used to contact the remote server.

**START-SESSION OPERATION.** START-SESSION validates the correspondent identity and right to communicate with TRANSFER at the local node, and establishes the running environment for a new session. This UOW cannot be issued from within an established session; TRANSFER only performs work for one session within a single IPC request. If present in a request, this UOW can be preceded only by NOOP or GET-CONFIG-NAME UOWs.

When inserting the START-SESSION UOW in an IPC request to initiate a session, you should set the SESSION-ID field of the IPC header to binary zeros; as a result, TRANSFER does not use this field for session validation.
If the session is successfully established, it will be effective for all later UOWs in the same request; TRANSFER returns the ID for the newly created session in the reply IPC header for that request. You must then set the SESSION-ID field to this value in all subsequent requests associated with that session.

NOTE

TRANSFER automatically eliminates a session that remains idle for an extended period of time. The controlling parameter, IDLESESSIONDELAY, in the PATHWAY configuration file has a default value of 24 hours.

To terminate the session, issue the END-SESSION UOW.

In COBOL or SCREEN COBOL, the following code sequence can be used to set the local time field of this UOW:

```
01 COBOL-TIME.
   05 TRANSFER-TIME PIC 9(6).
   05 FILLER PIC 9(2).

PROCEDURE DIVISION.

   MOVE 19 TO CENTURY.
   ACCEPT ACCEPT-DATE FROM DATE.
   ACCEPT COBOL-TIME FROM TIME.
   MOVE ACCEPT-TIME TO TRANSFER-TIME.
```
UOW Descriptions
SUBMIT-PKG

SUBMIT-PKG (UOW Code 117)

SUBMIT-PKG submits a package for delivery.

```
DEF submit-pkg-uow.
  02 hdr.
    03 self-ident
    03 uow-code PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 117.
  02 item-id.
    03 dummy PIC X(12).
END.
```

```
DEF submit-pkg-rsp.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 117.
  02 retn-code
    02 retn-code-detail TYPE BINARY 16.
END.
```

SUBMIT-PKG FIELDS. The fields in this UOW are:

- HDR is the UOW header. The UOW-CODE value is 117.
- ITEM-ID identifies the package header item for the package to be submitted.
- RETN-CODE is the return code. TISERV returns a code in this field to indicate one of the following entries:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OK</td>
</tr>
<tr>
<td>4010</td>
<td>E-BAD-TRANSACTION</td>
</tr>
<tr>
<td>4035</td>
<td>E-ITEM-NOT-FOUND</td>
</tr>
<tr>
<td>4036</td>
<td>E-ITEM-TOO-COMPLEX</td>
</tr>
<tr>
<td>4040</td>
<td>E-BAD-ITEM-DESCR</td>
</tr>
<tr>
<td>4041</td>
<td>E-ITEM-UNALTERABLE</td>
</tr>
<tr>
<td>4042</td>
<td>E-ITEM-NOT-PKG-HDR</td>
</tr>
<tr>
<td>4043</td>
<td>E-PREVIOUSLY-SUBMITTED</td>
</tr>
<tr>
<td>4045</td>
<td>E-TSCHED-UNAVAIL</td>
</tr>
<tr>
<td>4075</td>
<td>W-TIME-WINDOW-EXTENDED</td>
</tr>
<tr>
<td>4077</td>
<td>E-LIFESPAN-TOO-LONG</td>
</tr>
<tr>
<td>4079</td>
<td>W-PRIORITY-REDUCED</td>
</tr>
<tr>
<td>4082</td>
<td>E-NO-RECIIPS</td>
</tr>
<tr>
<td>4083</td>
<td>E-NOT-CREATED-BY-YOU</td>
</tr>
</tbody>
</table>
• RETN-CODE-DETAIL is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

SUBMIT-PKG OPERATION. SUBMIT-PKG submits a previously created package for delivery. A package can only be submitted by its creator.

If the submittal is successful, TRANSFER marks the package and all of its component items as unalterable.

When the package expiration time is reached, TRANSFER automatically removes the package from the INBOX folders of all recipients who have not acknowledged receipt.

You can cancel delivery of a submitted package by issuing the CANCEL-PKG UOW.

If you wish to change the delivery parameters (such as delivery window or priority) for a package, you must issue the ALTER-ITEM-DESCR UOW prior to the SUBMIT-PKG UOW.
UNSAVE-ITEM (UOW Code 111)

UNSAVE-ITEM removes an item from a folder.

**UNSAVE-ITEM FIELDS.** The fields defined in this UOW are:

- **HDR** is the UOW header. The UOW-CODE value is 111.
- **ITEM-ID** identifies the item to be removed from the folder.
- **FOLDER-NAME** is the name of the folder from which the item is to be removed.
- **RETN-CODE** is the return code. TISERV returns a code in this field to indicate one of the following.

  To indicate successful unsaving of the item:

  0 OK
To indicate problems with the folder name:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5675</td>
<td>E-FLD-NSRV-ERR</td>
</tr>
<tr>
<td>5676</td>
<td>E-FLD-NOT-FOUND</td>
</tr>
<tr>
<td>5677</td>
<td>E-FLD-BAD-NAME</td>
</tr>
<tr>
<td>5678</td>
<td>E-FLD-BAD-TYPE</td>
</tr>
<tr>
<td>5679</td>
<td>E-FLD-NO-SUCH-NODE</td>
</tr>
<tr>
<td>5680</td>
<td>E-FLD-SECURITY</td>
</tr>
<tr>
<td>5681</td>
<td>E-FLD-NSRV-NOT-FOUND</td>
</tr>
<tr>
<td>5682</td>
<td>E-FLD-NSRV-DOWN</td>
</tr>
<tr>
<td>5683</td>
<td>E-FLD-AMBIGUOUS-NAME</td>
</tr>
<tr>
<td>5684</td>
<td>E-FLD-SECURITY</td>
</tr>
</tbody>
</table>

To indicate other problems:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4010</td>
<td>E-BAD-TRANSACTION</td>
</tr>
<tr>
<td>4035</td>
<td>E-ITEM-NOT-FOUND</td>
</tr>
<tr>
<td>4061</td>
<td>W-ITEM-NOT-IN-FOLDER</td>
</tr>
<tr>
<td>4105</td>
<td>E-CONCURRNT-FLD-UPDATE</td>
</tr>
</tbody>
</table>

- RETN-CODE-DETAIL is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

UNSAVE-ITEM OPERATION. UNSAVE-ITEM removes the item identified by ITEM-ID from the folder identified by FOLDER-NAME.

Prior to any UNSAVE-ITEM request for an item, you can issue a SAVE-ITEM request to save the item in the WASTEBASKET folder. This will allow the application to maintain access to the item and reclaim it should the need arise at any time before ending the current session.

WARNING

When an item is unsaved from a folder, the item is scheduled for deletion from the system. The item descriptor and associated data will be deleted if three conditions are satisfied: (1) the item is not saved in any folder at the particular node, (2) the item is not the component of any other item, and (3) the item is a package and the delivery window has passed.

If the transaction containing an UNSAVE-ITEM UOW is aborted after the UNSAVE is completed, it is possible that the item being unsaved will have its item descriptor and associated data deleted but the item entry will still be referenced in the folder. As a result, subsequent SCAN-FOLDER UOWs for the folder will return the item ID; but UOWs requiring access to the item descriptor or data will result in an ITEM-NOT-FOUND error code.
UOW Descriptions
UNSAVE-ITEM

To resolve the UNSAVE problem, you can save the item in the WASTEBASKET folder before unsaving it from the desired folder; issue the SAVE-ITEM in the WASTEBASKET in the same TMF transaction as the UNSAVE-ITEM. This prevents the item descriptor and associated data from being deleted because the item is now saved in another folder. The item will be automatically purged from the WASTEBASKET and deleted from the item descriptor and associated data when the current session is ended.
WHERE-SAVED (UOW Code 121)

WHERE-SAVED retrieves the names of folders containing the specified item.

```
DEF where-saved-UOW.
  02  hdr.
    03  self-ident  PIC AA VALUE "UW".
    03  uow-code   TYPE BINARY 16 UNSIGNED VALUE 121.
  02  item-id.
    03  dummy      PIC X(12).
    02  prev-folder PIC X(80).
  02  num-requested TYPE BINARY 16 UNSIGNED VALUE 5.
END.
```

```
DEF where-saved-rsp.
  02  hdr.
    03  self-ident  PIC AA VALUE "UW".
    03  uow-code   TYPE BINARY 16 UNSIGNED VALUE 121.
  02  retn-code   TYPE BINARY 16.
  02  retn-code-detail TYPE BINARY 16.
  02  num-returned TYPE BINARY 16 UNSIGNED.
  02  foldersreturned PIC X(80) OCCURS 0 TO !num-requested! 5 TIMES DEPENDING ON num-returned.
END.
```

WHERE-SAVED FIELDS. The fields defined in this UOW are:

- HDR is the UOW header. The UOW-CODE value is 121.
- ITEM-ID identifies the item.
- PREV-FOLDER is the folder name following which retrieval is to start. You can begin with the first folder by setting this field to binary zeros.
UOW Descriptions
WHERE-SAVED

- NUM-REQUESTED is the number of folder names to be retrieved. This value directly determines the length of the response returned by TISERV.

- RETN-CODE is the return code. TISERV returns a code in this field to indicate one of the following entries.

  To indicate successful retrieval:

  0  OK

  To indicate problem with the folder name:

  5675  E-FLD-NSRV-ERR
  5676  E-FLD-NOT-FOUND
  5677  E-FLD-BAD-NAME
  5678  E-FLD-BAD-TYPE
  5679  E-FLD-NO-SUCH-NODE
  5680  E-FLD-SECURITY
  5681  E-FLD-NSRV-NOT-FOUND
  5682  E-FLD-NSRV-DOWN
  5686  E-FLD-SECURITY

  To indicate other problems:

  4001  W-EOF
  4035  E-ITEM-NOT-FOUND
  4092  E-INVALID-NUM-RQSTD
  4105  E-CONCURRNT-FLD-UPDATE

- RETN-CODE-DETAIL is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

- NUM-RETURNED is the number of folder names returned by TISERV in the response.

- FOLDERS-RETURNED is an array that contains the folder names returned by TISERV. The size of this array must be consistent with the value specified in NUM-REQUESTED. The array begins with the folder that follows the folder specified in PREV-FOLDER and that contains the item.

NOTE

Since TISERV retrieves the folders in order of their internal identification, the folder names are not arranged in any readily apparent sequence in this array.
WHERE-SAVED OPERATION. WHERE-SAVED returns, in the array named FOLDERS-RETURNED, the names of folders that contain the item identified by ITEM-ID. The UOW begins retrieval with the next folder that follows the folder denoted by PREV-FOLDER, and obtains as many folder names as the number indicated by NUM-REQUESTED. Only folders within the depot represented by the current session are considered.

The number of folders returned might be smaller than the number requested because there are no more folders that contain the item. In this case, the W-EOF warning is returned in the RETN-CODE field.
WRITE-PROFILE-REC (UOW Code 213)

WRITE-PROFILE-REC writes or updates a user-maintained depot profile record.

```
DEF write-profile-rec-uow.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 213.
  02 corr-name PIC X(80) VALUE SPACES.
  02 rec-type PIC 9(4) COMP.
  02 rec-seq-num PIC 9(4) COMP.
  02 depot-flag TYPE BOOLEAN VALUE "Y".
  02 filler TYPE CHARACTER 1.
  02 update-control PIC S9(4) COMP.
  02 num-data TYPE BINARY 16 UNSIGNED.
  02 profile-data.
    03 byte OCCURS 0 TO 1000 TIMES DEPENDING ON num-data PIC X.
END.
```

```
DEF write-profile-rec-rsp.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code TYPE BINARY 16 UNSIGNED VALUE 213.
  02 retn-code TYPE BINARY 16.
  02 retn-code-detail TYPE BINARY 16.
  02 corr-name PIC X(80).
END.
```

WRITE-PROFILE-REC FIELDS. The fields defined in this UOW are:

- HDR is the UOW Header. The UOW-CODE value is 213.

- CORR-NAME is the name of the correspondent for whose depot the profile record is to be written. This is also the name of the correspondent represented by the calling process. Only system administrators can enter correspondent names other than their own in this field; correspondents without system administrator privileges must enter their own names or leave the field blank. On input, you can use a partially qualified name, including wildcard characters. In the response, the fully qualified name is returned.
• **REC-TYPE** is the type assigned to the record by your application. The value in this field must be greater than 799.

  Values 800 through 999 - reserved for use by Tandem
  Values 1000 through 9999 - available for customer use

• **REC-SEQ-NUM** is the sequence number to be assigned to the record. The number can range from 1 through 9999.

• **DEPOT-FLAG** determines whether the UOW references a depot profile record or a system control record.
  
  Y = a depot profile record, as indicated by CORR-NAME
  N = a system control record

  If you make no entry in this field, Y is assumed.

A depot profile record contains information specific to a particular depot. When a new depot is created, all profile records in a model depot are copied into the new depot.

A system control record contains global parameters for the node. There can be a unique system control record for any particular REC-TYPE/REC-SEQ-NUM for each node at which your TRANSFER system is running. This record survives deletion of depots, and is not copied into new depots when they are created. Use of information in these records to supply default values for corresponding depot control parameters is determined entirely by your application.

• **UPDATE-CONTROL** is the counter used to control concurrent update of the profile record between read and write operations. This is the update number that TRANSFER writes into the record each time an update takes place. You should obtain this value through the READ-PROF-REC-A02 UOW, and then pass it back to TRANSFER on the call to the WRITE-PROFILE-REC UOW. If another process has updated the record since the calling process read it, an error message E-UPDATE-MISMATCH is returned.

  - To add a new record, set this field to zero.
  - To bypass this test and update the record regardless of whether anyone else has updated it, set this field to -1.

• **NUM-DATA** is the number of bytes written by TISERV. This entry can have a maximum value of 1500.
UOW Descriptions
WRITE-PROFILE-REC

• PROFILE-DATA is the array that contains the new record to be written, or the update information for an existing record. You can modify the array size within the system limitation of 1500 bytes.

• RETN-CODE is the return code. TISERV returns a code in this field to indicate one of the following entries.

To indicate successful record writing or updating:

0 OK

To indicate problems with the correspondent name:

5600 E-CORR-NSRV-ERR  5607 E-CORR-NSRV-DOWN
5601 E-CORR-NOT-FOUND   5611 E-CORR-NET-DOWN
5602 E-CORR-BAD-NAME     5622 E-CORR-NOT-SAME-NODE
5604 E-CORR-NO-SUCH-NODE 5623 E-CORR-AMBIGUOUS-NAME
5606 E-CORR-NSRV-NOT-FOUND

To indicate other problems:

4010 E-BAD-TRANSACTION  4051 E-MUST-BE-YN
4046 E-INVALID-REC-TYPE 4058 E-INVALID-REC-SEQ-NUM
4047 E-REC-NOT-FOUND    4093 E-SECURITY-VIOLATION
4049 E-REC-ALREADY-EXISTS 4201 E-CONTEXT-ERR
4050 E-UPDATE-MISMATCH  4902 E-ERR-PROFILE-FILE

• RETN-CODE-DETAIL is an error number returned by a subsystem other than TRANSFER or is a further qualification of an error detected by TRANSFER.

WRITE-PROFILE-REC OPERATION. WRITE-PROFILE-REC writes or updates a user-maintained record in the depot Profile file. The UOW must start on a word boundary. Each response consumes a whole number of words of the IPC reply.

To update a record, you supply the update counter for that record in the UPDATE-CONTROL field. When you issue the WRITE-PROFILE-REC UOW with a nonzero value in the UPDATE-CONTROL field, TRANSFER first reads the old record (locking it temporarily) and compares the UPDATE-CONTROL value in the record with that which you have supplied; the old record must already exist.
If the two values match, or if the value you have supplied is -1:

The record is updated with the new information supplied in PROFILE-DATA.

The updated record is written (and unlocked) with its old UPDATE-CONTROL value incremented by 1. TRANSFER ensures that this value wraps around from 9999 to 1 every 10000 updates.

To write a new record, set the UPDATE-CONTROL field to zero. TRANSFER takes the input from PROFILE-DATA, and writes it into a new record in the file. If the record indicated by REC-TYPE and REC-SEQ-NUM already exists, TRANSFER returns an error condition.
SECTION 6
DEVELOPING TRANSFER APPLICATIONS

An application can be designed and developed in many different ways. As a general guideline, Tandem suggests the following combination of steps as one possible way to proceed:

1. Decide whether your application should use TRANSFER.

2. Specify the functional aspects of the application. Identify correspondents and what they do, what kind of information they exchange, how they use that information, and how they acknowledge receipt of packages.

3. Design input screens for users at terminals.

4. Divide the application tasks among clients, agents, and other application programs.

5. Define the package formats and protocol for communication among correspondents.

6. Plan high-level transactions, taking TMF requirements into consideration.

7. Plan low-level implementation.

8. Code and test the software.

APPLICATION DEVELOPMENT STEPS

The application development steps presented in this section are only guidelines that suggest an approach; they do not imply design prerequisites or a rigid methodology for developing a TRANSFER application. Statements about what different entities, such as requesters and servers, should do are simply ideas that offer one clear model of how TRANSFER can be used.
Developing TRANSFER Applications

Step 1: Deciding Whether Your Application Should Use TRANSFER

Consider using TRANSFER if your application requires delivery of information to multiple correspondents, particularly across a network, or if it depends upon the staging of transactions over time. These requirements need not be the primary thrust of the application; even if they are only a small part of it, TRANSFER can still help you considerably.

TRANSFER is not an application in itself; it delivers information (packages), but does not process them. TRANSFER does not inherently allow you to defer the transport of packages that have already been posted, nor does it guarantee their delivery in any particular sequence.

TRANSFER emphasizes reliability of delivery rather than speed of transmittal. TRANSFER cannot guarantee that a package will arrive at a precise time, nor can it predict the exact amount of time that a package will spend in transit. This makes TRANSFER unsuitable for time-critical applications such as real-time communication where one terminal operator must converse online with another.

Step 2: Specifying the Functional Aspects of the Application

Identify the correspondents that send and receive packages. Consider whether these correspondents are people, processes, devices, or other entities; your application must interface with each of these types differently.

Consider how many recipients your application will involve, and how they might be grouped into distribution lists. Determine whether all recipients receive identical packages. Decide what action a recipient of a package takes, such as adding items or forwarding it to another recipient.

Define the characteristics of the packages that correspondents exchange. Consider such factors as:

- What kind of information does the package contain? TRANSFER is indifferent to package content. TRANSFER allows applications to define packages, items, and record types for their own purposes; and to nest packages and items inside one another.

- Is the order in which packages arrive at a depot important? Your application can embed sequencing information inside packages, and provide agent code to process packages in sequence.

- Is the time window in which packages will be delivered a consideration? This might involve distinguishing between recipients that are batch applications running only at night,
Developing TRANSFER Applications

and online applications active during the day. It might also require careful analysis of the daily availability of devices that are recipients of packages.

• How do the correspondents use this information?

• Do the correspondents need to acknowledge package receipt? Some packages might demand a reply from the recipient to the sender.

After considering these factors, define the various transactions that will involve extended (nowait) processing. Such transactions normally begin when a correspondent composes and sends a package. The end of the transaction, however, depends on your application. For example, you can define a transaction to end:

• when an agent at the last recipient's depot receives notification that the package arrived

• when the last intended recipient saves the package in a folder or deletes it

• when the sender receives notification for every recipient that the package was received, or notification that the package could not be delivered or that it expired before it was received

• when every recipient has taken some action and explicitly replied to the sender

• when the sender has taken some specific action based on the reply.

When replies to senders are part of a transaction, define exactly what the reply means to the sender and how that reply is linked to the original package. Examples of deliveries and replies that constitute complete transactions appear in Figure 6-1.

A receiving correspondent can reply to a sender in any of the following ways:

• acknowledge the delivery without reading the package

• acknowledge the delivery after reading the package, but without performing any other processing

• acknowledge the delivery after fully processing the package.

Decide whether to require certification of package delivery. Consider whether return packages will be more useful or more annoying to the sender, and whether the need for certification warrants the additional package traffic.
Figure 6-1. Complete Transactions Involving Extended Processing
Developing TRANSFER Applications

Consider the level of data protection and integrity that your application must provide. This might involve the use of fault-tolerant features, TMF, or both.

Try to anticipate any operational and administrative requirements that your application imposes upon others at your site.

Step 3: Designing Input Screens for Users at Terminals

If your application involves users at terminals, design the input screens and establish conventions for reporting errors.

Determine what screens are needed, how the screens should be grouped, and what path the user should take while progressing through particular screen sequences. As an example:

- A client presents a user with a log-on screen.
- After the user logs on, the client produces a menu screen from which the user can select various groups of functions, such as compose a package.
- After a group is selected, the client allows the user to choose a specific function, such as add a new item or modify an existing one.

Examples of screens presented by a client appear in Section 7.

Screen design should be consistent. You can provide consistency by selecting a standard screen layout. You could divide each screen in the application into four basic areas that always appear in the same locations on each screen:

- screen identification and headings
- screen body
- an area for special instructions to the operator
- an area for error message reporting.

The example in Figure 6-2 shows the T/Mail application screen for displaying the contents of an INBOX folder.
Developing TRANSFER Applications

Carefully integrate the use of function keys with the screen displays. An operator will use these keys either to signal completion of an operator function or to specify a choice, such as add an item or delete an item. To avoid errors and confusion:

- Use the keys consistently on all screens.
- Lay out the references to the keys distinctly on the screen.
- Limit the number of function keys needed on each screen.
- Avoid using multiple keys for the same function.
- Confine screen entries to similar, related choices.
- Consider using shifted keys for functions that might produce irreversible results.
Developing TRANSFER Applications

- Provide a series of help screens and menu screens to help the user to progress through the application.

- Distinguish various kinds of fields in the screen layout. You might use underlining or reverse video for entry fields, dim display for special prompts, and blinking or brightened display for error messages.

- Use default values and protect crucial screen fields to help prevent errors. SCREEN COBOL field characteristic clauses can be used to specify integrity constraints, and your application can perform interfield checking after screen editing is complete.

If you are adding TRANSFER to an existing application, decide how to integrate the new functions, screens, and keys into the existing application.

Step 4: Dividing Application Tasks Among Clients, Agents, and Other Programs

Consider how the work performed by the application should be divided among clients, agents, and other application programs. In making these decisions, refer to the discussions of clients and agents in this manual, and to the PATHWAY manuals referenced in the preface.

Step 5: Defining Package Formats and Protocol for Communication Among Correspondents

Design the structure of the packages and replies, and determine how correspondents should interpret them. Often, a single application uses many package formats. Designing a package entails not only deciding what the structure will be, but also defining delivery parameters such as priority, certification, and timeframes.

When you include one package within another, the number of levels of nesting can affect performance. To retrieve the components on each level, a requester must:

1. retrieve the components of the outermost package by issuing a request to TISERV

2. if any of the components is itself a package, retrieve its components by issuing another request

3. continue this process until all components have been retrieved at all levels.
Developing TRANSFER Applications

Retrieving the nested components of a package requires at least one SEND statement for every level of nesting in the package. Multiple SEND operations, however, degrade the response time of the application.

The impact of nesting is even more significant upon asynchronous requesters.

Requesters must retrieve all components in order to transport a package to each remote node.

Each level of nesting results not only in a separate SEND operation to transmit the data at each level, but also in a recursive call to the program unit that transports the data.

Deeply nested packages can exhaust the data space available to an asynchronous requester and degrade the performance of TRANSFER across nodes.

An alternative to nesting packages is to build them horizontally, not vertically. Figure 6-3 illustrates two structures that have the same number of component items, but those items are spread more evenly across the second structure than across the first:

Retrieving the components of the first structure requires six SEND operations, including one SEND to get the item ID of the outermost package.

Retrieving the components of the second structure requires three SEND operations. If the package needs to be transported across nodes, the second structure entails less recursive processing by asynchronous requesters.
Figure 6-3. Avoiding a Nesting Problem
Developing TRANSFER Applications

Step 6: Planning High-Level Transactions

A TRANSFER transaction typically consists of several TMF transactions, possibly in different PATHWAY systems. Often, separate TMF transactions are required to build and post the package; deliver the package; respond to the package, such as placing it in a folder on the receiving end; produce and post replies; deliver the replies; and react to the replies. All of these transactions, except the ones involving package delivery, are in the domain of application requesters and agents.

For simple applications, it is generally sufficient to:

- Start a TMF transaction immediately after completion of a terminal read for any request that requires a transaction.
- End this TMF transaction immediately before the next terminal read.

The following recommendations deserve special attention:

- A TMF transaction should never span a read operation from a terminal. This promotes concurrent accesses to the data base.
  - If you allowed a TMF transaction to span a terminal read, the record locks involved in the transaction might be held for a very long time, delaying access by other users.
  - TMF does not allow the audit trail records for a single transaction to span more than a certain number of audit trail files, as determined by a TMFCOM parameter; thus, if a transaction takes so long that several audit trail file switches occur, TMF aborts the transaction.
- Updates which, in combination, transform the data base from one consistent state to another consistent state, should be grouped into a single TMF transaction; otherwise, you fail to take advantage of the TMF guarantee that all or none of the changes in a single transaction are maintained in the data base. Avoid grouping together too many of these transformations because the record locks involved are held for the life of the transaction; a lengthy transaction might maintain locks long enough to dramatically reduce the amount of work on the data base that could proceed concurrently.
- Requests that do not involve changes to files on disc do not require a TMF transaction.
- You only need multiple TMF transactions per user request in cases where a single request might update hundreds of records.

For additional information on TMF transaction processing, refer to the PATHWAY SCREEN COBOL Reference Manual.
Step 7: Planning Low-Level Implementation

Low-level implementation decisions involve packaging your application in individual modules (COBOL or SCREEN COBOL programs, FORTRAN subroutines, or TAL procedures) and subroutines, breaking the application up into small groups of functions. They also involve carefully planning the interfaces between these modules and subroutines, and deciding what information is passed between them.

You can plan for various accounting procedures, collecting usage statistics, and the logging of particular events and errors. Initially, you might want to log every SEND operation requested by a SCREEN COBOL program to provide extensive debugging information.

If a large group of programmers is working on the application, plan and establish coding standards for the group. These standards prevent implementation conflicts, recoding, and readjustment as the development phase progresses.

Plan the methods for reporting errors to users. Your application can find out about errors in several different ways, as illustrated in Figure 6-4.

1. Problem reports can arrive as packages in the INBOX folder. A package might state that another package expired before it was acknowledged or could not be delivered on time to a specific recipient.

2. Errors in a request, but not specific to a UOW, are reported by TRANSFER in the reply header. These errors usually indicate programming errors in the SCREEN COBOL program that makes the request. Examples of such errors are an incorrect session ID or an incorrect request length (more or fewer UOWs than the request indicated).

3. Errors encountered while processing specific UOWs are reported in the response UOWs by TRANSFER. Examples of such errors are ITEM-NOT-FOUND, for a UOW that refers to an item, or INVALID-FOLDER-NAME, for a UOW that names a folder. An operator could have entered an invalid name, which the requester passed to the server as a parameter in a UOW.

In some cases, an error might be cause to abort and restart a transaction. If a transaction is creating an item and adding records and the request fails because a processor fails, you cannot tell how many records were added successfully to the item; in this case, you should abort and restart the transaction.

4. Errors can be reported by the GUARDIAN operating system, input/output processes, and other software in response to file system requests if the application includes its own
servers. PATHWAY reports a variety of errors to SCREEN COBOL programs.

5 Certain errors are reported to the system administrator as console messages or entries to a log file.

Figure 6-4. Ways in Which Errors Are Reported
Error handling strategies vary with the type of error and with the application. For example, an agent might react to reports of expiration or delivery problems by reporting those conditions to the operator, revising a set of values used to calculate timeframes and priorities, or automatically posting a package again under certain conditions. Figure 6-5 shows an application that maintains parameters for calculating timeframes; if too many of a correspondent's packages cannot be delivered on time, the agent that notices the problem changes the parameters in the application profile.

Step 8: Coding and Testing the Software

Before coding your programs, read the associated language and PATHWAY manuals. Consider carefully the conventions for ensuring fault-tolerant operation and for using TMF to guarantee data base consistency. Various support notes and Tandem Softdoc information might be useful to you.

NOTE

For programs written in TAL or FORTRAN, values entered in fields with data types equivalent to the COBOL/SCREEN COBOL type PIC(4) COMP are limited to the range 0 through 9999. This limitation applies even if it is not explicitly stated in the TAL or FORTRAN definition for fields of that particular type. Although a PIC(4) COMP field would be assigned type INT in a TAL program, entry of a value less than 0 or greater than 9999 in that field would still result in an error (such as E-INVALID-REC-TYPE).

Fields with the PIC X characteristic, such as the ITEM-ID or SESSION-ID fields, are not displayable as output to terminal sessions.

Test your application module-by-module. If your application is based upon the requester/server model, you can test your servers alone before evaluating their interaction with SCREEN COBOL requesters. Later, you can evaluate your requesters with debugged servers.
Developing TRANSFER Applications

Figure 6-5. An Agent Monitors Package Delivery

1. The operator posts a package.
2. The requester gets the timeframe from the profile.
3. Sets the delivery parameters.
4. And submits the package.
5. The agent sees that too few packages are being delivered successfully.
6. And changes the constant for calculating timeframes.
Developing TRANSFER Applications

When your application is running, you should validate its fault-tolerant operation and reliance upon TMF. This might entail a recovery drill where operators use TMF to recover from various kinds of failures.

DESIGNING AND WRITING A CLIENT

Every TRANSFER application must include one or more clients. Clients are the programs that allow correspondents using the application to communicate and interact with TRANSFER. In most cases, the main job of the client is to provide an interface between correspondents who are users at terminals and the TRANSFER software.

Clients let correspondents issue requests for building, altering, and posting packages; and for receiving, reading, and responding to incoming packages. Clients also let correspondents request various administrative functions, such as defining new correspondents and distribution lists. The application forwards these requests to TRANSFER by issuing UOWs to the TRANSFER interactive server (TISERV). TRANSFER then responds by performing the operations requested.

A client can be either a stand-alone client that operates as an entire application, or a dependent client that is actually a component of a larger application.

Examples of stand-alone clients are:

- An electronic mail system, such as the T/MAIL application supplied by Tandem.
- An online central filing system where packages are deposited by some correspondents to be read by others.
- A suspense file application that reminds correspondents of events scheduled to take place on certain days. With this type of application, correspondents might scan a folder for Thursday, July 29th, and find packages alerting them to various meetings scheduled for that day; or the application might use delayed delivery to automatically alert correspondents to imminent events whenever they log on.

Examples of dependent clients are:

- Applications not originally designed to use TRANSFER. In such applications, TRANSFER might be almost invisible to the users. An example is a warehousing application where the user presses one function key to review an invoice and another to transmit the invoice to a remote node.
Developing TRANSFER Applications

• Applications involving T/MAIL. In some cases, the client might invoke T/MAIL to perform various functions, usually by means of menu-screen entries. In other cases, T/MAIL might call the client to perform some specialized function such as formatting special interoffice memos or invoices. Specific information about clients that interact with T/MAIL appears in Appendix B.

Most clients are written in SCREEN COBOL and execute in PATHWAY Terminal Control Processes (TCPs); others are written in COBOL, FORTRAN, or TAL. Only SCREEN COBOL clients, however, can take advantage of PATHWAY and its load-balancing features, ease of coding interprocess communications and TMF transactions, and resource management capabilities.

As you plan the design of your client, consider the following:

1. The client divides its attention among the correspondents; therefore, the total number of correspondents affects the time in which the system can respond to each.

2. If a client represents correspondents who are processes, those processes can be PATHWAY server classes, and possibly agents. If a correspondent is a nonstandard device, the process that issues requests to the device can be either a server (possibly an agent) or a terminal simulator. In any case, a server still requires a SCREEN COBOL program to make requests; each copy of a SCREEN COBOL program must be assigned either a physical terminal or a terminal simulator. Many of these applications might not require PATHWAY.

3. Any request on behalf of a correspondent must be directed to a server class running at the node where the correspondent is registered.

Grouping UOWs in IPCs

A single SEND statement or WRITEREAD call can request multiple services, as long as all of those services are for the same correspondent. There is only one request header to identify the correspondent. For each service requested, your application must issue one UOW.

In your program, you group UOWs into an interprocess communication (IPC) that is forwarded to TISERV by the SEND statement or WRITEREAD call. One IPC exists for each SEND statement or WRITEREAD call; one UOW exists for each operation requested. TRANSFER processes the UOWs in the order in which they appear in the IPC.
Developing TRANSFER Applications

The more UOWs you can pack into one IPC (SEND or WRITEREAD request) and the fewer IPCs in your program, the better. For applications running under PATHWAY, reducing the number of SEND operations between screen presentations is particularly important. Both TRANSFER and PATHWAY operate far more efficiently with fewer IPCs. Excessive communication slows the system down, and the processing overhead associated with each request further contributes to this slowdown.

Fewer IPCs offer several advantages:

- results in fewer input/output operations by TISERV as well as fewer interprocess messages
- minimizes the points of interaction between your client and TISERV, potentially simplifying error recovery
- provides an easy way for sequences of operations to control the execution of later operations if an early operation fails.

Packing as many UOWs into as few IPCs as possible is important. In extreme cases, it might even be feasible to start a session, perform all of the work for that session, and terminate the session, within the same IPC.

To facilitate the coding of several UOWs within the same IPC, you can use the RQST-CODE field of the IPC header to specify whether or not to continue processing all UOWs in an IPC in the event of an error or a warning regarding a particular UOW within that IPC. If the RQST-CODE field contains STOP-ON-ERROR, an error condition in a UOW prevents succeeding UOWs in the IPC from being processed. If the RQST-CODE field contains DO-ALL-UOWS, however, an error or a warning will not terminate processing of the remaining UOWs in the IPC.

If you abort a transaction because of an error in an IPC, all changes to the TRANSFER data base caused by this IPC are backed out.

If successful completion of one UOW is necessary for later UOWs, place the dependent UOWs in the same IPC and specify STOP-ON-ERROR.

Suppose, for example, you wish to discard an item in a folder, but before discarding it, you want to save it in the WASTEBASKET folder so that it will remain available until the end of the current session:

Save the item (with a SAVE-ITEM UOW) and then remove it from the originating folder (with the UNSAVE-ITEM UOW) in the same IPC.

If the SAVE-ITEM UOW fails, the UNSAVE-ITEM UOW will not even be attempted and you will still retain the item.
Developing TRANSFER Applications

By putting the SAVE-ITEM and UNSAVE-ITEM UOWs in the same IPC and transmitting the IPC within the framework of a TMF transaction, you can ensure that both of these operations take place or that neither of them do. If the SAVE-ITEM UOW succeeds but the UNSAVE-ITEM UOW fails, TMF will remove the item from the WASTEBASKET folder when the transaction is aborted.

Multiple UOWs in an IPC can complicate error handling:

- You must look at each UOW individually.
- You cannot direct TRANSFER to use the results returned by earlier UOWs as input to later UOWs in the same IPC.
- You must issue two or more IPCs, possibly moving the earlier results to the later requests in the client code.

NOTE

The total size of an IPC or its reply cannot exceed the maximum value defined at TISERV startup time, as indicated in Table 4-1 of Section 4. For PATHWAY requesters, this value is specified at PATHWAY configuration time as indicated in the TRANSFER Delivery System Management and Administration Guide.

Communicating with TISERV

A general rule for PATHWAY applications is to divide server classes so that each request sent to each server takes approximately the same amount of time. When directing UOWs to TISERV, however, you cannot consider individual operations; you must think of groups of UOWs.

Although this makes it more difficult to estimate the time required to handle the UOWs, it does give you more flexibility in balancing the workload per request among servers. One possible approach to this situation might be to define several server classes, with each class using the same server program but different response-time characteristics.
Using Correspondent Names for Distribution Lists

You can use correspondent names as an aid in naming distribution lists. As an example, you could create a correspondent called LIST to help identify public distribution lists. This would indicate to accessors that they were indeed accessing a distribution list; it would also help them recall the correct way to reference the list by requiring them to remember only the distribution list name. Examples would be:

LIST.SALESMAN
LIST.MAILER @LA

TMF Guidelines

When using TMF, consider the following guidelines:

1. Multiple TMF transactions can significantly impact program execution time; therefore, consider performing all of the work for each request within a single TMF transaction—even if your application does not require the guarantee that either all or none of the individual operations will be completed. In general, your program will only require multiple TMF transactions for a single user request if the request will take a long time, or if it involves updates to hundreds of records in the data base.

2. Changes requested in a TMF transaction are not reflected in the data base until the TMF transaction ends; therefore, applications should not leave a TMF transaction active while waiting for a user to press a function key or react to a prompt. The record locks involved in such a transaction might be held for a very long time, delaying access by other users. Also, if the user leaves the terminal with a TMF transaction still in progress and an error occurs, audit trails on disc cannot be purged.

3. Your application should abort TMF transactions whenever a server detects an error that would make the data base inconsistent. An example of such a TMF transaction is one in which one item is being attached to, or detached from, another. This involves changing both the PARENT-COUNT of the component item and the COMPNT-COUNT of the parent item; an error occurring between these changes would leave the TRANSFER data base inconsistent.
NOTE

Always abort a transaction when a data file is full or when any error that your program is not prepared to handle is returned on a file.

Always abort a transaction when the BAD-TRANSACTION message is returned. This indicates a fatal error, such as the transaction itself no longer existing in the system, a recipient's node going out of service, or some type of system failure.

Network Guidelines

When your application issues a START-SESSION UOW for a correspondent, it must direct this UOW to a server or server class running at the node where the correspondent is registered. Your application must continue to make requests of that same server or server class because the SESSION-ID assigned in the response to the START-SESSION UOW is defined only at the same node.

SCREEN COBOL Coding Guidelines

Guidelines for coding SCREEN COBOL programs operating in a PATHWAY environment appear in the PATHWAY SCREEN COBOL Reference Manual. Guidelines for coding programs in COBOL, FORTRAN, or TAL (which do not run under PATHWAY) appear in the manuals covering those languages. Additional rules apply specifically to programs that are TRANSFER clients:

1. Always abort any transaction if the PW-REPLY-CODE field in the IPC header is set to a value greater than 1 (indicating an error in the IPC header or in an individual UOW). This will retain the consistency of your data base and make it easier to debug your application and get it running again.

2. When coding a client, mistakes creating inconsistencies between the amount of buffer space provided for a reply in an IPC and the actual length of the data returned are not uncommon. You can trap this kind of error by including an ON-ERROR clause in each SEND statement that transmits UOWs.

3. To help conserve working storage, some UOWs are set up so that you can use the response from one as the request portion in another. For example, the response to the GET-PROFILE-ELEM UOW can be used as a request in the ALTER-PROFILE-ELEM UOW. Similarly, the response from the GET-AGENT-SELECT UOW can be used as a request in the ALTER-AGENT-SELECT UOW.
Developing TRANSFER Applications

4. The working storage established for the initial transmission of some UOWs to TISERV can be used repeatedly in subsequent transmissions of the same UOW. This is true, in particular, where the responses for some UOWs have the same format and can thus be used as the requests. The READ-NEXT-MEMBER and READ-NEXT-NAME UOWs fall into this category. Such UOWs also return parameters properly for the next call, so you need not move data around when taking advantage of this feature.

Debugging Techniques

As an option, TRANSFER will record on the debug log file information that passes between your client and TISERV. To enable this logging, you issue an ASSIGN directive and two PARAM directives to the server, as illustrated by this example:

```
ALTER SERVER serverclass, (ASSIGN DEBUGLOG, $vol.subvol.file),&
(PARAM DEBUGLOGLEVEL 0), (PARAM DEBUGLOGFORMAT TRUE)
```

- The ASSIGN directive specifies the log file, which is the logical file DEBUGLOG.
- The first PARAM directive selects which information is logged, as follows:

<table>
<thead>
<tr>
<th>DEBUGLOGLEVEL</th>
<th>The following action results:</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>All IPCs are logged.</td>
</tr>
<tr>
<td>1</td>
<td>IPCs containing UOW warnings, UOW errors, and IPC request errors are logged.</td>
</tr>
<tr>
<td>2</td>
<td>IPCs containing UOW errors and IPC request errors are logged.</td>
</tr>
<tr>
<td>3</td>
<td>IPCs containing IPC request errors are logged. This option is assigned as a default.</td>
</tr>
<tr>
<td>4</td>
<td>Nothing is logged.</td>
</tr>
</tbody>
</table>

...
Developing TRANSFER Applications

- The second PARAM directive determines the format in which the log file is written. Normally, this file is written as a binary file; but through this PARAM directive, you can direct TRANSFER to write it in displayable text by setting DEBUGLOGFORMAT to TRUE.

  Alternatively, you can use the GUARDIAN operating system File Utility Program (FUP) to convert the binary content of the log file to ASCII, octal, or hexadecimal code.

Additional logging techniques include:

  Logging on a depot-by-depot basis - Use the ALTER-PROFILE-ELEM UOW to set the DATA-DEPOT-LOG-FLAGS field to request this option.

  Logging selectively - Choose particular IPCs for recording on the log file. Set the LOG-THIS-IPC field to Y in the headers of those IPCs that you want logged. This action results in logging of those IPCs whether or not errors occur.

  Add information to the log file - Include in each IPC a NOOP UOW that specifies client data identifying the source of the transaction requested in the IPC.

The TRANSFER Scheduler (TSCHED) and the TRANSFER asynchronous requesters (TAREQs) write messages to the OUT file used by TSCHED. These messages reflect problems dealing with initialization, UOWs, IPC headers, and SEND and HOLD operations. If, in addition to the OUT file, you have configured a TSCHED log file, you can examine this log file for possible additional debugging information.

DESIGNING AND WRITING AN AGENT

A TRANSFER application frequently includes one or more agents. These agents are notified when TRANSFER delivers a package to a recipient's depot, and generally take some kind of action as a result. Agent notification is handled by the TAREQ that delivers the package. In response, a particular agent can perform tasks such as filing the package for later retrieval or replying to the sender of the package.

An agent eliminates the need for polling to see if packages have arrived. An agent also allows the application to automatically process incoming packages without human intervention.
The precise responsibilities of agents vary considerably. Some agents retrieve and save packages for their correspondents, delete packages that are not meaningful to their recipients, present messages on output devices, or initiate transactions based upon package delivery. Specific examples are:

- A vacation agent that takes some kind of action in the absence of a package recipient. This agent might reply automatically to all packages; forward the packages to another correspondent, optionally appending text that provides additional information or directives; file the packages for later retrieval; or delete items according to some criteria in the package header.

- A filtering agent that screens incoming mail according to certain criteria; for example, filing packages that pertain to certain subjects or sorting packages according to type before presenting them to a recipient. The agent might place mail memos in one folder and reply messages in another. The agent might file packages from different applications separately; then, the client can allow the correspondent to retrieve the packages selectively.

- An agent that arranges in proper sequence packages that are required to arrive in a certain order. To use an agent in this way, the application would establish an arrangement such as the following:

  1. The sender embeds sequence numbers in the package. As an example, the sequence numbers might be included as data records in the package headers.

  2. The agent at the recipient's depot processes those packages that arrive in the expected order, but saves in a folder any packages that arrive out of order. The folder then becomes a queue for out-of-sequence packages.

  3. Whenever a package has been processed, the agent checks the folder for a package with the next higher sequence number. If the folder contains such a package, the agent processes that package next.

- An interfacing agent that supports the transmittal of packages between a Tandem network and a network consisting of systems furnished by other manufacturers. The sender of the mail, at a Tandem system node, specifies the external mail address of the recipient in a suffix appended to the recipient's name. The agent then uses the suffix to address the package, entering it for delivery in the other system. The application might, in fact, use the interchange formats currently recommended by the National Bureau of Standards as a basis for these operations.
Developing TRANSFER Applications

Agents can operate as either requesters or servers.

- An agent that makes requests of TRANSFER server classes to retrieve, save, or post packages is a requester written in SCREEN COBOL. In this role, an agent can be regarded as a specialized client.

- An agent that accesses a GUARDIAN file or input/output device, performs string-handling or complex computations, or does work of long duration, is typically a server written in COBOL, FORTRAN, or TAL.

- An agent that performs both kinds of work is generally written in SCREEN COBOL and makes requests of a special server provided by your application.

SCREEN COBOL agents run in the same TCP as the TAREQs that invoke them. The communication between agent and TAREQ is program unit to program unit, and never TCP to TCP.

The program unit for the TAREQ is located in the TAREQ SCREEN COBOL object code files (TAREQCOD and TAREQDIR). Locate a SCREEN COBOL agent in one of the following ways:

Compile the SCREEN COBOL agent and copy the program unit into the TAREQ SCREEN COBOL object code files by using the SCREEN COBOL Utility Program (SCUP). Refer to the PATHWAY SCREEN COBOL Reference Manual for a description of SCUP commands.

or

Compile the SCREEN COBOL agent and locate the program unit in a SCREEN COBOL object code file other than the one in which TAREQ resides. You can then supply the PATHCOM TERM TCLPROG parameter for the TAREQ and specify your agent program unit object file as the TCLPROG filename. Refer to the PATHWAY System Management Reference Manual.

An agent required by a package delivery is invoked only if the delivery is successful. The same agent can be associated with several depots, but no depot is required to have an agent. Conversely, many agents can be associated with the same depot.
Developing TRANSFER Applications

TAREQs invoke agents in accordance with selection criteria recorded in the recipient's agent profile. Agent selection criteria can be established interactively through the ADMIN client provided by TRANSFER, or programmatically through the ALTER-AGENT-SELECT UOW issued to TISERV. The sender can use fields within package headers to specify information that must match the selection criteria in the recipient's profile in order for the agent to be executed. This makes it possible for different agents at a depot to handle different types of packages and applications. More information about agent selection and the addition of agents to depots appears in the discussion of the ALTER-AGENT-SELECT UOW.

Assigning Names to Agents

If an agent is to run as a PATHWAY requester, you must assign it a valid SCREEN COBOL program name. The name:

- can contain from 1 through 30 characters
- can include only alphanumeric characters or hyphens
- must not include embedded blanks
- must not begin or end with a hyphen.

If an agent is to run as a PATHWAY server, you must assign it a valid server name. The name:

- can contain from 1 through 15 characters
- can include only alphanumeric characters or hyphens
- must begin with a letter
- must be unique within the PATHWAY system in which it is defined
- must not include embedded blanks
- must not be a PATHWAY reserved word.

Any agent with a name longer than 23 characters cannot take advantage of the ADMIN client's user escape sequences for agent configuration and deletion programs. These programs are described in the Configuring or Deleting an Agent paragraph.
Developing TRANSFER Applications

NOTE

When processing an agent name, TISERV verifies that the name contains no embedded blanks. TISERV, however, does not enforce the more restrictive rules and does not verify that the agent really exists in the TRANSFER system.

Establishing Sessions for Agents

If an agent needs access to a correspondent's depot, it must either establish a session or be granted one. Through the ALTER-AGENT-SELECT UOW, you can set up the correspondent's profile so that the agent is automatically granted a session whenever it is invoked.

When an application attempts to start a session for a correspondent, TRANSFER checks to ensure that another session is not already in progress for the same correspondent. Sessions created for agents, however, are not counted in this verification. Thus, the E-CONCURRENT-SESSION or W-CONCURRENT-SESSION message is only issued if other non-agent sessions are active for the correspondent. If an agent that requires a session is configured at a depot where concurrent sessions are forbidden, the agent does not fail if invoked while the depot has an active session.

TAREQ Interface

When a TAREQ invokes a SCREEN COBOL agent, the TAREQ must wait until the agent returns; when a TAREQ invokes a server class agent, the TAREQ must wait until the agent replies. While the agent is performing its task, the TAREQ that invoked the agent remains busy with that agent and can perform no other functions. Therefore, the more work an agent does, the more TAREQs your application will require to handle the workload. This is particularly true where many packages are each frequently delivered to many recipients; the TAREQ must always perform a unique delivery for every recipient.

A SCREEN COBOL agent is expected to perform all of its work within the framework of the TMF transaction provided by the TAREQ.
Developing TRANSFER Applications

WARNING

A SCREEN COBOL agent must not issue a BEGIN-TRANSACTION, ABORT-TRANSACTION, or END-TRANSACTION. A SCREEN COBOL agent can only issue a RESTART-TRANSACTION.

If a SCREEN COBOL agent issues a RESTART-TRANSACTION, TMF backs out the delivery as well as all work performed by preceding agents within the scope of the TAREQ TMF transaction. Then the TAREQ retries the transaction for the number of times specified by the MAXTMFRESTARTS configuration parameter in PATHWAY.

If the SCREEN COBOL agent persists in restarting the transaction, the current function that the TAREQ is processing is placed on HOLD. HOLD always means that the TAREQ will make no further attempt to deliver the package locally for the duration of the HOLD. HOLD might also mean that an expiration function will not be scheduled, and the package might not be transported to other nodes for the duration of the HOLD. The function remains in the scheduler READY file, but is not subject to selection until it is set to UNHOLD by starting the scheduler with PARAM UNHOLD TRUE.

If an agent is a requester written in SCREEN COBOL, the TAREQs invoke it with CALL statements. If an agent belongs to a PATHWAY server class, TAREQs address it by issuing SEND statements to the class.

For additional information, refer to the TAREQ Interface discussion in Section 4 and the description of the ALTER-AGENT-SELECT UOW in Section 5.

Selecting a Programming Language

An agent that requires ongoing access to the TRANSFER data base should be written in SCREEN COBOL.

An agent that frequently accesses only data bases that are not part of TRANSFER, or is not heavily involved in any data base access, should be written in COBOL, FORTRAN, or TAL; thus, the TCP will not be involved in this agent's activity and will be free for other operations. The invoking TAREQ, however, will not be free because it is devoted to the agent.
Developing TRANSFER Applications

Debugging an Agent

When debugging an agent, you can develop a program unit that calls the agent and displays a screen that allows you to pass information to the agent, thus simulating a TAREQ. This lets you observe more directly the interaction between the TAREQ and agent. If this program is written as a user mail extension, it will even more closely simulate the TAREQ environment. User mail extensions are discussed in Appendix B.

To use the system DEBUG facilities, you can define the server class with the DEBUG option of the ADD SERVER or SET SERVER PATHCOM command set to ON.

Configuring or Deleting an Agent

When you configure an agent, you must supply certain information for the agent in the depot profile record. This information is described in the discussions of the ALTER-AGENT-SELECT and GET-AGENT-SELECT UOWs. To enable an operator to supply this information, Tandem provides a standard (default) configuration program that is sufficient for many cases.

You can supply your own agent configuration program and tailor it to the specific needs of your agent. The program can display a screen through which the operator can enter information; the program can next determine which of the entries are appropriate for copying to the profile record and then issue the ALTER-AGENT-SELECT UOW to write the information to that record.

The AGENT-DATA field, as modified and retrieved by the ALTER-AGENT-SELECT and GET-AGENT-SELECT UOWs, respectively, can contain up to 80 characters of data. The Default Agent Configuration Screen displayed by the standard default configuration program allows the viewing and modification of only the first 78 characters of this field. If you are providing an agent that requires more than 78 characters of agent data, you must furnish your own configuration program for the agent.

When deleting an agent, the default action of the ADMIN application is not sufficient if your agent configuration module produces special information that is not stored as part of the agent selection criteria in the agent profile. In this case, you must provide your own agent deletion program to delete this information.

Your agent configuration and deletion programs must be written in SCREEN COBOL.
### INVOKING THE PROGRAMS

The configuration or deletion program is invoked when an operator presses one of the function keys on the ADMIN Agent Selection Screen. If TRANSFER does not locate a user-supplied program, the following occurs:

- The default program named ADMIN-AGENT-CONFIGURE, which is supplied by Tandem, is invoked in response to the configuration function keys.
- The ADMIN default action takes place in response to the delete function key.

If TRANSFER locates a user-supplied program, the program is invoked and takes the action dictated by the function key:

<table>
<thead>
<tr>
<th>Function Key</th>
<th>Program/Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>FL3</td>
<td>Invokes your SCREEN COBOL configuration program to configure the agent named on the screen, and then inserts the name of this agent in the list of agents for the depot. TRANSFER obtains the name of the SCREEN COBOL program run unit by appending -CONFIG to the supplied agent name.</td>
</tr>
<tr>
<td>F8</td>
<td>Invokes your SCREEN COBOL configuration program to reconfigure the agent named on the screen, which already appears in the list of agents for the depot. TRANSFER obtains the name of the SCREEN COBOL program run unit by appending -CONFIG to the supplied agent name.</td>
</tr>
<tr>
<td>SF13</td>
<td>Invokes your SCREEN COBOL deletion program to delete the agent named on the screen. TRANSFER obtains the name of the SCREEN COBOL program run unit by appending -DELETE to the supplied name.</td>
</tr>
</tbody>
</table>

### CALLING THE CONFIGURATION OR DELETE PROGRAM

The ADMIN application calls the configuration program through the following calling sequence, where xxxxxxxxxx is the name of the agent supplied on the screen:

```call xxxxxxxxxx-CONFIG USING LNKSC-IPC-HDR-L, LNKSD-LINKAGE-BLOCK, LNKSE-ERROR-BLOCK, LNKSF-AGENT-MANAGE-DATA-L.```
Developing TRANSFER Applications

The ADMIN application calls the deletion program through this calling sequence:

    CALL xxxxxxxx-DELETE USING LNKSC-IPC-HDR-L,
        LNKSD-LINKAGE-BLOCK,
        LNKSE-ERROR-BLOCK,
        LNKSF-AGENT-MANAGE-DATA-L.

Generally, your SCREEN COBOL program has the option of completing the requested configuration or deletion operation itself, refusing to do the operation, or doing part of the operation and then requesting TRANSFER to do the rest.

The parameters passed to your program by the call are blocks of data that must be defined in the Linkage Section of the SCREEN COBOL module. Any restrictions on updating the fields on the screens displayed by a user-supplied module must be defined in the user-supplied module. The parameters are as follows:

- **LNKSC-IPC-HDR-L** is the standard IPC header. If your program issues a UOW, the program must modify the REQUEST-CODE and UOWS-TO-PROCESS fields. Do not alter other fields, or unpredictable results might occur in ADMIN after your program returns.

- **LNKSD-LINKAGE-BLOCK** contains general linkage information used between modules. The DDL definition is:

  ```
  DEF linkage-block.
      02 corr-name TYPE *.
      02 sys-admin-flag TYPE CHARACTER 1.
      02 tmf-started-flag PIC 9.
      02 system-name PIC X(8).
      02 pathmon-name PIC X(8).
      02 return-state TYPE BINARY 16.
      02 tiserv-server PIC X(16).
      02 admin-server PIC X(16).
      02 filler PIC X(208).
  END.
  ```

In this definition, the fields have the following meanings:

- **CORR-NAME** is the correspondent name of the logged-on correspondent, or the correspondent name entered on the ADMIN Main Menu Screen by a system administrator. Do not alter this field.

- **SYS-ADMIN-FLAG** indicates whether the logged-on correspondent is a system administrator. Do not alter this flag.
Developing TRANSFER Applications

TMF-STARTED-FLAG is used to control error recovery if your SCREEN COBOL program terminates abnormally. You must set this field to 1 immediately before beginning a TMF transaction, and set it to 0 immediately after terminating a TMF transaction with END-TRANSACTION or ABORT-TRANSACTION.

SYSTEM-NAME is the name of the EXPAND node at which the correspondent named in the CORR-NAME field is registered. Do not alter this field.

PATHMON-NAME is the name of the PATHMON process running the TRANSFER system at the node named in the SYSTEM-NAME field. Do not alter this field.

NOTE

A future release of TRANSFER will allow the EXPAND node name and the PATHMON process name in the SYSTEM-NAME and PATHMON-NAME fields to be different from those names under which your user-supplied module is running. To ensure compatibility with future versions of TRANSFER, you must specify these two field names in the network form of the SCREEN COBOL SEND statement.

RETURN-STATE determines where your program returns; set this field to 0 to return to the calling module, or 1 to return to the ADMIN Main Menu Screen. Do not enter any other values in this field.

TISERV-SERVER and ADMIN-SERVER are server class names for TISERV and ADMIN. Do not alter these fields.

FILLER is unused, and is reserved for use by Tandem.

- LNKSE-ERROR-BLOCK can be used to return an error message to the calling screen. Generally, you should only use this field if your SCREEN COBOL program does not provide for display of this message on a screen. The DDL definition is:

```cobol
DEF error-block.
  02 error-message TYPE *.
  02 error-attr-flag PIC 9.
  05 filler TYPE CHARACTER 13.
END.
```
Developing TRANSFER Applications

In this definition, the fields have the following meanings:

ERROR-MESSAGE contains the error message text, which can include up to 67 characters. The definition for this field is:

    DEF error-message.
      02 error-text TYPE CHARACTER 67 VALUE SPACES.
      02 error-detail.
        03 sign-field-2 TYPE CHARACTER 1 VALUE SPACES.
        03 error-field-2 PIC 9999 VALUE 0.
        03 sign-field-1 TYPE CHARACTER 1 VALUE SPACES.
        03 error-field-1 PIC 9999 VALUE 0.
        03 unusable TYPE CHARACTER 3 VALUE SPACES.
    END.

ERROR-ATTR-FLAG specifies the kind of video display in which the error message appears. To display the message in inverse video, set this field to 1. To display the message in normal video, set this field to 0.

FILLER is unused.

• LNKSF-AGENT-MANAGE-DATA-L contains a return action flag and data for an ALTER-AGENT-SELECT UOW. The DDL definition is:

    DEF agent-manage-data.
      02 return-action-flag PIC S9(4) COMP.
      02 agent-alter-data TYPE GET-AGENT-SELECT-RSP.
    END.

In this definition, the fields have the following meanings:

RETURN-ACTION-FLAG contains the result of your program's execution. It can be set to one of the following values.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>Your program rejected the configuration or deletion request, either by intent or because of an error in any UOW or server request it attempted. (The program might have placed a message in the error block.) The TMF-STARTED-FLAG and RETURN-STATE fields should both be set to 0.</td>
</tr>
<tr>
<td>0</td>
<td>Your program completed the entire operation successfully and requires no action by TRANSFER. This implies that your program issued the appropriate UOWs itself. The TMF-STARTED-FLAG should be set to 0.</td>
</tr>
</tbody>
</table>
### Value | Meaning
---|---
1 | Your program requested TRANSFER to issue the ALTER-AGENT-SELECT UOW. If your program sets the TMF-STARTED-FLAG, TMF will honor the status of that flag. (Your program can do this if it writes data to another file and then requires TRANSFER to complete the agent selection operation under the same transaction ID that your program obtained.) The RETURN-STATE field should be set to 0.

AGENT-ALTER-DATA, on entry, contains the data for an ALTER-AGENT-SELECT UOW appropriate for the add, alter, or delete agent operation being performed. The fields either contain the current values, or default to new values. The user-supplied module can modify these values as appropriate and then either issue the UOW itself, or return with RETURN-ACTION-FLAG set to 1 (in which case ADMIN will issue the modified UOW).

### USER-SUPPLIED PROFILE RECORDS AND MODULES

In addition to the standard profile records created and managed by TRANSFER, your application can allow the writing of user-supplied profile records to the Profile file. These are records whose structure, format, and content are defined by your application.

The ADMIN application provides predefined escape functions so that your application can create and manage these records through user-supplied profile modules. These are SCREEN COBOL programs that you write and provide for this purpose.

To invoke a module, an operator presses the user-supplied profile (F9) function key on the ADMIN application Main Menu Screen. The module then typically presents a screen that allows the operator to display and update the user-supplied profile records. Any restrictions on the use of this screen are defined by the module.

The ADMIN application calls the user-supplied profile module through the following calling sequence:

```
CALL ADMIN-PROFILE-USER USING LNKSC-IPC-HDR-L,
     LNKSD-LINKAGE-BLOCK,
     LNKSE-ERROR-BLOCK.
```
Developing TRANSFER Applications

The parameters passed by this call to the user-supplied module are the same as the first three parameters in the calls that invoke agent configuration or deletion programs. The general meanings of these parameters are described in the Configuring or Deleting an Agent paragraph.

USER-SUPPLIED SYSTEM CONTROL MODULES

A user-supplied system control module provides the means to set parameters for controlling the operating environment of your TRANSFER applications. This module, which displays one or more screens, is called by the ADMIN application when the SF9 function key is pressed from the Main Menu Screen. This SCREEN COBOL module must be named ADMIN-SYSTEM-USER. The calling sequence for the user-supplied system control module is:

CALL ADMIN-SYSTEM-USER USING LNKSC-IPC-HDR-L,
     LNKSD-LINKAGE-BLOCK,
     LNKSE-ERROR-BLOCK.

The parameters passed by this call to the user-supplied module are the same as the first three parameters in the calls that invoke agent configuration or deletion programs. The general meanings of these parameters are described in the Configuring or Deleting an Agent paragraph.

TMF TRANSACTION PROCESSING GUIDELINES

The basic guidelines for establishing TMF transactions in all programs apply to clients and agents. These are described in the Transaction Monitoring Facility (TMF) User's Guide. In addition to this information, however, you should also be aware of the following points that relate specifically to TRANSFER applications:

1. If the TRANSFER data base is audited exactly as recommended in the TRANSFER Delivery System Management and Administration Guide, TRANSFER makes the following guarantees.
TRANSFER will deliver every successfully submitted package within the time window specified for the package, or will notify the sender of its failure to deliver that package. TRANSFER will deliver a package only once to each recipient. If a failure to deliver a package occurs, TRANSFER will deliver its failure notification only once.

If a TMF rollforward operation is performed and the program that rebuilds the TRANSFER scheduler queues is run following a catastrophic (multipoint) failure, all submit and cancel operations in progress at the time of the failure will be performed again to completion.

All active (usable) data in the TRANSFER data base will remain consistent at all times.

2. While servicing a request, TRANSFER might perform many input/output operations on the TRANSFER data base. Under TMF transactions, these updates require record locking to ensure data base consistency and integrity. The maximum number of locks that can be acquired for each partition of a file in any data base is limited. Transactions might be affected by the lock limits of TMF in some cases.

To prevent difficulties imposed by these locking limits, keep the number of records per item, items per folder, recipients per distribution list, or recipients per package to moderate levels. Because TRANSFER blocks records rather than storing each record individually, these TMF locking limits do not always correspond exactly to the number of records per item, items per folder, and so forth actually permitted in a particular case. In the worst case, the values as specified by the TMF locking restrictions apply. In many other cases, the following rules might be helpful; these rules apply only if the data is added to the objects sequentially.

The amount of data in the item data records added to a single item should total less than 500,000 bytes. As an example, 6,200 80-byte data records per single item are permissible.

The number of recipients per package, or number of members per distribution list, should be less than 5,000.

The number of items per folder should be less than 500.

These values are overall limits that will allow TRANSFER to delete the objects involved later without exceeding the TMF locking limits.
Developing TRANSFER Applications

3. If your system manager has not arranged for TMF auditing exactly as recommended in the Transaction Monitoring Facility (TMF) User's Guide, additional considerations in dealing with files expected to be audited should be noted. For example, if the SESSIONS file in the TRANSFER data base is not audited, aborting a transaction that includes the START-SESSION or END-SESSION UOW might leave extraneous data in the data base. This could result in a WASTEBASKET folder that does not belong to any current session. As another example, if the ITEM-DATA file is not audited, item data records that no longer belong to any particular item might remain in the system after the item has been deleted.

ITEM AND PACKAGE DESIGN ALTERNATIVES

Certain constraints are imposed upon the depth to which you can nest component items within parent items or items within packages. If you have a choice within the requirements of your application, apply the following guideline:

It is better to structure large packages and items so that they contain many components at the same level but incorporate fewer levels, than to include an extreme number of levels with fewer components at each level.

TISERV Guidelines

When assembling and submitting a package under TISERV, the amount of nesting allowed can be limited by the amount of stack space available in the server for these activities. The space available is determined by the MEM option of the RUN command, and the MAXREQUEST, MAXREPLY, and MAXLINKS params defined when TISERV is started.

In spite of this constraint, a very deep item tree can be constructed by working from the top down. You would, for example, start with the items at the topmost level, link these items together, and then proceed to the second level, linking the items at this level together, and so forth.

TISERV checks the nesting when your application attaches a component item to a parent (with the ATTACH-COMPNT-A01 UOW), and examines the item tree to see if there is a reference to the parent anywhere in the item tree below the component. If there is such a reference, a component cycle exists as shown in Figure 6-6, and the error E-COMPNT-CYCLE is returned. If the item is nested too deeply, TISERV cannot continue checking for cycles and returns the error E-ITEM-TOO-COMPLEX.
TISERV also checks the nesting of an entire package when your application submits it for delivery (with the SUBMIT-PKG UOW). During this activity, TISERV must flag all items in the package as UNALTERABLE because the same package might be physically shared among several recipients. As in package assembly, if TISERV finds that the package is nested too deeply, it returns the error E-ITEM-TOO-COMPLEX.

TAREQ Guidelines

In addition to the limits imposed by TISERV, additional constraints might be imposed by TWORK, a special server used by TAREQs. When a package is being transported over the network, TWORK saves information recursively about each level of nesting, and the nesting depth permitted depends on the amount of stack space available in TWORK.
Developing TRANSFER Applications

You cannot alter the amount of stack space available in TWORK; stack space is already set to its maximum. TWORK stack usage per level of nesting is dependent on many things that are difficult to predict and are subject to change from one release to another. You can assume that any release will always support ten levels of nesting. If a package cannot be transported because of excessive nesting, DELIV-ERR in the recipient record of each remote recipient is set to E-ITEM-TOO-COMPLEX; and a TAREQ event package for the event PACKAGE CANNOT BE TRANSPORTED TO RECIPIENT'S SYSTEM is generated.

Addition of Records to Package Header Items

In addition to adding data records to component items, your application can also add them to package header items. When transmitting short messages, the application can economize by adding records directly to the package header rather than by creating and attaching other items to the header.

Both T/MAIL packages and TAREQ event packages can contain text records in lieu of components in their package headers.

NETWORK GUIDELINES

The following special considerations apply when you are using TRANSFER to deliver packages over a multi-node network:

1. Be sure that TMF is configured at all nodes in the network where TRANSFER is running.

2. A misspelling of network node designators within recipient names might not cause immediate rejection of a package. When your application specifies a node in a recipient name and remote name resolution is deferred, TRANSFER assumes that the node exists, places the package in the network queue, and waits for availability of the node. Although TRANSFER returns an appropriate warning in the case of a nonexistent node, the package is only withdrawn from posting by direct cancellation or by expiration of the delivery time window.

   If remote name resolution is not deferred, specifying an incorrect node name immediately results in an error.

3. If you defer remote name resolution when submitting a package, the package can be reproduced at the receiving node but with invalid recipients. When TRANSFER at that node detects such a recipient, it returns an error to the sending depot.
GENERATING THE SAMPLE CLIENT

The following steps explain how to configure and run the sample client that appears in Section 7.

Arbitrary names are shown in lowercase letters; substitute appropriate names for your installation.

1. Log on and set your current volume to the $volume.subvolume where XBEGIN resides.

2. The sample client source code has been supplied by Tandem and is on your system. The file is named SMPCLNT.

3. Compile the source code:

   `SCOBOL/IN SMPCLNT,OUT output-file/EXAMP;SYMBOLS`

4. Add these PATHCOM commands for your PATHWAY configuration:

   ```
   RESET PROGRAM
   SET PROGRAM TCP your-tcp
   SET PROGRAM TMF ON
   ADD PROGRAM SAMPLE-CLIENT, TYPE your-terminal-type &
   (INITIAL SAMPLE-CLIENT, TCLPROG EXAMP)
   ```

   **NOTE**

   These commands can be entered into the GBEPATH file that was created by XBEGIN, or they can be added interactively once your system is up and running.

5. Enter this PATHCOM command: `RUN SAMPLE-CLIENT`

6. The sample ELEC-MAIL Client logon screen appears.

7. Have the system administrator register, without passwords, your name and various other fictitious names. You can then log on as yourself, or as these other correspondents, and send packages back and forth through the TRANSFER system.
SECTION 7
SAMPLE CLIENT

This section presents an example of the operations performed by a client. In this example, the client allows the correspondent to:

1. initiate interaction with the application by logging on
2. assemble packages and transmit them to recipients
3. receive and read packages from other senders
4. terminate interaction with the application by logging off.

Despite its simplicity, this client illustrates some of the most important functions available with TRANSFER, shows how SCREEN COBOL screens provide interaction between a client and a correspondent, and demonstrates how and when several commonly used UOWs are issued. These UOWs include:

- `ACK-RECEIPT`
- `ADD-ITEM-REC`
- `ADD-RECIP`
- `CREATE-ITEM`
- `END-SESSION`
- `GET-ITEM-DESCR`
- `GET-ITEM-REC`
- `SCAN-FOLDER`
- `START-SESSION`
- `SUBMIT-PKG`
- `UNSAVE-ITEM`

FUNCTIONAL DESCRIPTION OF THE SAMPLE CLIENT

After the sample client is started by the System Manager and presents its Logon Screen, the correspondent at a terminal interacts with the client as follows:

1. The Logon Screen includes two entry fields: a 32-character field for the correspondent name and a 16-character field for the password, as shown in Figure 7-1. After entering these fields, the correspondent presses the F1 function key to transmit them to the client. The SF16 function key allows the correspondent to exit the program.
Sample Client

2. The client passes the information from the Logon Screen to the input buffer for the START-SESSION UOW, executes a BEGIN-TRANSACTION operation to denote the beginning of a TMF transaction, and then issues an IPC containing the START-SESSION UOW to TISERV. If the UOW is successful, TISERV assigns a session ID and begins a session for the correspondent; the client now executes an END-TRANSACTION operation to terminate the transaction and displays its Function Menu Screen. If the UOW fails, TISERV reports the failure to the client, which aborts the transaction and displays an error message on the Logon Screen.

When creating the session, TISERV returns the session ID in the SESSION-ID field of the IPC header. (Previously, this field contained binary zeros.) A client must enter this same session ID value in the headers for all subsequent IPCs associated with this session.
3. The Function Menu Screen, shown in Figure 7-2, allows the correspondent to select one of four functions by pressing the appropriate function key: read mail, send mail, log off and return to the previous screen, or log off and exit the program. The correspondent, deciding to compose and send some mail, presses the F5 key.

![Function Menu Screen](image)

**Figure 7-2. Client's Function Menu Screen**

4. In response to the F5 function key, the client next displays its Mail Posting Screen, as shown in Figure 7-3. This screen allows the correspondent to enter the name of the recipient of the package, a subject line, and up to five lines of text. Upon entering this information, the correspondent presses the F6 key to post the package. The F16 function key would allow the correspondent to stop package composition and return to the Function Menu Screen.
To handle the package, the client executes a BEGIN-TRANSACTION operation and issues an IPC containing a CREATE-ITEM UOW to establish a package header item. This package header item is characterized by the data in the ITEM-TYPE and IS-PKG-HDR fields of the UOW. In the reply to this UOW, TISERV returns a unique item ID for the new item.

As part of the same transaction, the client issues:

a. another IPC that contains:

1) an ADD-RECIPI UOW to define the recipient of the package

2) a group of ADD-ITEM-REC UOWs to add the subject and text lines to the package. This group can include from 0 to 1 ADD-ITEM-REC UOWs for the subject line, and from 0 to 5 ADD-ITEM-REC UOWs for the text line. (See the ADD-SUBJECT-REC and ADD-TEXT-REC definitions in the code presented for this example at the end of this section.)
Sample Client

b. still another IPC that contains:

1) an ADD-ITEM-REC UOW to add the recipient name
   returned in the ACCEPTED-NAME field of the reply to
   the ADD-RECIP UOW transmitted in Step 4-a-1. This
   step is taken so that the client will produce
   packages that are compatible with TRANSFER/MAIL
   (T/MAIL), which expects to find its recipient as text
   in the IPC header.

   2) a SUBMIT-PKG UOW to submit the package for delivery.

At this point, the client executes an END-TRANSACTION
operation to terminate the transaction, and TRANSFER assumes
control of the package to deliver it asynchronously. Now,
the client once more displays its Function Menu Screen.

5. Next, the correspondent decides to read incoming mail.
Through the Function Menu Screen, the correspondent selects
this operation by pressing the F3 function key.

In response, the client issues an IPC containing the
SCAN-FOLDER UOW, referencing the correspondent's INBOX
folder. This UOW returns the IDs of the items in the folder
to the client, reading them in sequential order. Notice that
since this IPC did not involve changes to the TRANSFER data
base (and thus, did not involve possible inconsistencies in
the event of an error), this IPC was not issued within the
framework of a TMF transaction.

Now, for the first package in the list of item IDs, the
client executes a BEGIN-TRANSACTION operation and sends an
IPC containing the following UOWs:

a. a GET-ITEM-DESCR UOW to obtain the item's descriptor
   fields

b. a GET-ITEM-REC UOW to obtain the subject record from the
   item

c. a GET-ITEM-REC UOW to obtain the recipient for the
   package

d. a GET-ITEM-REC UOW to obtain the text records

e. an ACK-RECEIPT UOW to acknowledge that the package was
   received at the recipient's depot.

Next, the client executes an END-TRANSACTION operation.
(Note that the presence of the ACK-RECEIPT UOW required that
the above IPC be contained within its own transaction.)
Sample Client

Now, the client displays the date and time that the package was posted, the sender's and recipient's names, and the subject and text records on the Mail Display Screen as shown in Figure 7-4. At this point, the client waits for the correspondent to press another function key to read the next package or discard it, or to return to a previous screen.

If TISERV returns W-EOF and no items in the reply to the SCAN-FOLDER UOW, the client displays an error message on the screen.

![Mail Display Screen]

Figure 7-4. Client's Mail Display Screen

6. The correspondent decides to read the next package in the INBOX folder, and presses the F3 function key. If there are more entries in the list, the client reissues the data-collecting UOWs in Steps 5-a through 5-e. If no more items appear in the list and TISERV has not informed the client that there are also no more in the folder, then the client returns to the SCAN-FOLDER UOW at the beginning of Step 5 and continues reissuing the sequence of UOWs from that point. If TISERV indicates that the list is empty, the client again displays the Function Menu Screen.
7. The correspondent can discard the package from INBOX while reading the mail by pressing the F4 function key. In response, the client executes a BEGIN- TRANSACTION operation, issues an UNSAVE-ITEM UOW, and then executes an END- TRANSACTION operation. (If the UNSAVE UOW fails, TISERV reports the error and the client displays an error message.) If more packages reside in INBOX, TISERV displays the next package.

8. After reading the mail, the correspondent presses the F16 key to return to the Logon Screen and log off. In response, the client begins another transaction and issues the END-SESSION UOW. When TISERV terminates the session, the client ends the transaction and again presents the Logon Screen. (At this point, TISERV also resets the SESSION-ID field in the IPC header to binary zeros.)

RUNNING THE SAMPLE CLIENT

The SCREEN COBOL code required to support this simple client appears in Figure 7-5, and is explained with comments embedded in the code. In this code, you will see:

- UOWs that are issued outside the framework of a transaction
- IPCs with one or more UOWs comprising the total content of a transaction
- transactions that span more than one SEND operation.

The code also illustrates how to define your own copies of definitions supplied by TRANSFER, and then modify the lengths of fields or number of occurrences specified by some of them. (This is done, for instance, in the ADD-ITEM-REC, ADD-SUBJECT-REC, and SCAN-FOLDER definitions in this example.)

These definitions can be copied from the GCOB and GLNK files into a program during compilation, as illustrated by the 01 record level definitions in this example. The GCOB file contains COBOL source text for elements commonly used in TRANSFER programs, such as IPC header and UOW definitions. The GLNK file contains the same text as the GCOB file, but without the INITIAL-VALUE clauses. The GLNK file is used for Linkage Section definitions.

Where feasible, the conventions established by T/Mail for REC-TYPE and other values are also applied in this example. This will allow T/Mail to handle packages created by this client.
Sample Client

The following steps explain how to configure and run the sample client.

Arbitrary names are shown in lowercase letters; substitute appropriate names for your installation.

1. Logon and set your current volume to the $volume.subvolume where XBEGIN resides.

2. The sample client source code has been supplied by Tandem and is on your system. The file is named SMPCLNT.

3. Compile the source code:

   SCOBOL/IN SMPCLNT,OUT output-file/EXAMP;SYMBOLS

4. Add these PATHCOM commands for your PATHWAY configuration:

   RESET PROGRAM
   SET PROGRAM TCP your-tcp
   SET PROGRAM TMF ON
   ADD PROGRAM SAMPLE-CLIENT,TYPE your-terminal-type &
   (INITIAL SAMPLE-CLIENT,TCLPROG EXAMP)

   NOTE

   These commands can be entered into the GBEPATH file that was created by XBEGIN, or they can be added interactively once your system is up and running.

5. Enter this PATHCOM command: RUN SAMPLE-CLIENT

6. The sample ELEC-MAIL Client logon screen appears.

7. Have the system administrator register, without passwords, your name and various other fictitious names. You can then logon as yourself, or as these other correspondents, and send packages back and forth through the TRANSFER system.
IDENTIFICATION DIVISION.

PROGRAM-ID. SAMPLE-CLIENT.

AUTHOR. TANDEM

INSTALLATION.

DATE-COMPILED. 84/09/17 - 17:36:28.

This program is an example client. Although simplistic in nature, it clearly illustrates the methods used to communicate with TRANSFER.

It is intended to show a simple interface to TRANSFER, not to define good SCREEN COBOL programming techniques.

This client assumes that:

1. The correspondent name entered on the logon screen must have been previously registered via the ADMIN function.

2. The standard INBOX is the ONLY folder used.

   This example allows the submission, retrieval, and deletion of packages submitted to TRANSFER. Where possible, the conventions established by TRANSFER/MAIL for REC-TYPE and other values will be used. This should allow T/MAIL to handle packages created by this example client.

   A base screen is used to display program-wide data such as headings and error displays. The individual functions are implemented via overlay screens. This decreases the amount of data sent to the terminal.

   The subject and text lines are written out as individual item records. This technique would allow an open-ended type of application (where the total number of records is unknown).
Figure 7-5. SCREEN COBOL Code for Client (continued)
Figure 7-5. SCREEN COBOL Code for Client (continued)
### Figure 7-5. SCREEN COBOL Code for Client (Continued)

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
<th>PIC</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>184</td>
<td>10 ih-IPC-RETN-CODE-DETAIL</td>
<td>S9(4)</td>
<td>COMP</td>
</tr>
<tr>
<td>185</td>
<td></td>
<td></td>
<td>VALUE</td>
</tr>
<tr>
<td>186</td>
<td>10 ih-SESSION-ID.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>187</td>
<td>15 ih-DUMMY</td>
<td>X(18)</td>
<td></td>
</tr>
<tr>
<td>188</td>
<td>10 ih-UOWS-TO-PROCESS</td>
<td>S9(4)</td>
<td>COMP</td>
</tr>
<tr>
<td>189</td>
<td>10 ih-UOWS-RETURNED</td>
<td>S9(4)</td>
<td>COMP</td>
</tr>
<tr>
<td>190</td>
<td></td>
<td></td>
<td>VALUE</td>
</tr>
<tr>
<td>191</td>
<td>10 ih-LOG-THIS-IPC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>192</td>
<td>10 FILLER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>95</td>
<td>*********</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Figure 7-5. SCREEN COBOL Code for Client (Continued)***

**PAGE 5 $MERC.B9110.SMPCLNT TANDEM / SCREEN COBOL**

---

```
96 /
97 ********** UOW DEFINITIONS. The DDL-generated definitions for the UOWs
98 * used in this program are copied into the WORKING-STORAGE Section *
99 * from the files named GCOB and GLNK, supplied by Tandem. *
100 ************************************************************
101
102 01 start-session-uow.
103 * COPY uow-ss-start-session-uow OF "GCOB".
104 2x 1805 * Definition START-SESSION-UOW created on 08/31/84 at 16:55
105 2x 1806 05 uow-ss-START-SESSION-UOW.
106 2x 1807 10 uow-ss-HDR.
107 2x 1808 15 uow-ss-SELF-IDENT PIC AA.
108 2x 1809 15 uow-ss-UOW-CODE PIC 9(4) COMP value 101.
109 2x 1810 10 uow-ss-APPLIC-ID PIC 9(4) COMP.
110 2x 1811 10 uow-ss-CORR-NAME PIC X(120).
111 2x 1812 10 uow-ss-PASSWORD PIC X(16).
112 2x 1813 10 uow-ss-CURR-LOCAL-TIME.
113 2x 1814 10 uow-ss-CORR-LOCAL-TIME.
114 2x 1815 15 uow-ss-YEAR PIC 9(4).
115 2x 1816 15 uow-ss-MONTH PIC 9(2).
116 2x 1817 15 uow-ss-DAY-OF-MONTH PIC 9(2).
117 2x 1818 15 uow-ss-HOUR PIC 9(2).
118 2x 1819 15 uow-ss-MINUTE PIC 9(2).
119 2x 1820 15 uow-ss-SECOND PIC 9(2).
120 2x 1821 10 uow-ss-REDEF-LOCAL-TIME REDEFINES uow-ss-CURR-LOCAL-TIME.
121 2x 1822 15 uow-ss-CENTURY PIC 9(2).
122 2x 1823 15 uow-ss-ACCEPT-DATE PIC 9(6).
123 2x 1824 15 uow-ss-ACCEPT-TIME PIC 9(6).
124
125
126 01 start-session-rsp.
127 * COPY rsp-ss-start-session-rsp OF "GCOB".
128 2x 1826 * Definition START-SESSION-RSP created on 08/31/84 at 16:55
129 2x 1827 05 rsp-ss-START-SESSION-RSP.
130 2x 1828 10 rsp-ss-HDR.
131 2x 1829 15 rsp-ss-SELF-IDENT PIC AA.
132 2x 1830 15 rsp-ss-UOW-CODE PIC 9(4) COMP.
133 2x 1831 10 rsp-ss-RETN-CODE PIC S9(4) COMP.
134 2x 1832 88 rsp-ss-OK VALUE 0.
135 2x 1833 88 rsp-ss-W-CONCURRENT-SESSION VALUE -4005.
136 2x 1834 88 rsp-ss-E-CONCURRENT-SESSION VALUE 4005.
137 2x 1835 88 rsp-ss-E-BAD-TRANSACTION VALUE 4010.
138 2x 1836 88 rsp-ss-E-NE-DEPOT-FOR-CORR VALUE 4011.
139 2x 1837 88 rsp-ss-E-RESTRICTED-OPERATION VALUE 4017.
140 2x 1838 88 rsp-ss-E-LOGON-DISALLOWED VALUE 4007.
141 2x 1839 88 rsp-ss-E-INVALID-PASSWORD VALUE 4015.
142 2x 1840 88 rsp-ss-E-INVALID-APPLIC-ID VALUE 4065.
143 2x 1841 88 rsp-ss-E-INVALID-DATE-TIME VALUE 4073.
144 2x 1842 88 rsp-ss-E-INVALID-ZONE-OFFSET VALUE 4021.
145 2x 1843 88 rsp-ss-E-ALREADY-IN-SESSION VALUE 4015.
146 2x 1844 88 rsp-ss-E-CONTEXT-ERR VALUE 4201.
147 2x 1845 88 rsp-ss-E-CORR-NOT-FOUND VALUE 5601.
148 2x 1846 88 rsp-ss-E-CORR-BAD-NAME VALUE 5602.
149 2x 1847 88 rsp-ss-E-CORR-BAD-SUFFIX VALUE 5624.
150 2x 1848 88 rsp-ss-E-CORR-BAD-TYPE VALUE 5603.
151 2x 1849 88 rsp-ss-E-CORR-NSRV-ERR VALUE 5600.
```

---
PAGE 6  $MERC.B9110.SMPCLNT  TANDEM / SCREEN COBOL

2X 1851  88  rsp-ss-E-CORR-SECURITY  VALUE 5605.
2X 1852  88  rsp-ss-E-CORR-NO-SUCH-NODE  VALUE 5604.
2X 1853  88  rsp-ss-E-CORR-NSRV-NOT-FOUND  VALUE 5606.
2X 1854  88  rsp-ss-E-CORR-NSRV-DOWN  VALUE 5607.
2X 1855  88  rsp-ss-E-CORR-NET-DOWN  VALUE 5611.
2X 1856  88  rsp-ss-E-CORR-AMBIGUOUS-NAME  VALUE 5623.
2X 1857  88  rsp-ss-E-CORR-NOT-SAME-NODE  VALUE 5622.
2X 1858  10  rsp-ss-RETN-CODE-DETAIL  PIC 59(4) COMP.
2X 1859  10  FILLER  PIC X.
2X 1860  10  rsp-ss-SERVER-VERSION.
2X 1861  15  rsp-ss-LETTER  PIC A value "B".
2X 1862  15  rsp-ss-REV-ZONE-DIFF  PIC 99 value 00.
2X 1863  10  rsp-ss-TIME-ZONE-DIFF  PIC S9(4) COMP.
2X 1864  10  rsp-ss-RESOLVED-NAME  PIC X(120).
01 end-session-uow.
10  * COPY uow-es-end-session-uow OF "GCOB".
2< 897  * Definition END-SESSION-UOW created on 08/31/84 at 16:48
2< 898  05 uow-es-END-SESSION-UOW.
2< 899    10 uow-es-HDR.
2< 900    15 uow-es-SELF-IDENT VALUE "UW".
2< 901    15 uow-es-UOW-CODE PIC 9(4) COMP value 102.
111
112 01 end-session-rsp.
113  * COPY rsp-es-end-session-rsp OF "GCOB".
2< 904  * Definition END-SESSION-RSP created on 08/31/84 at 16:48
2< 905  05 rsp-es-END-SESSION-RSP.
2< 906    10 rsp-es-HDR.
2< 907    15 rsp-es-SELF-IDENT VALUE "UW".
2< 908    15 rsp-es-UOW-CODE PIC 9(4) COMP.
2< 909    10 rsp-es-RETN-CODE PIC 9(4) COMP.
2< 910    88 rsp-es-OK VALUE 0.
2< 911    88 rsp-es-E-BAD-TRANSACTION VALUE 4010.
2< 912    88 (ROST-ERR if invalid session id)
2< 913    10 rsp-es-RETN-CODE-DETAIL PIC 9(4) COMP.
114 *********
The program must set ASU-DATA-BYTE-COUNT to 32, since a VALUE clause cannot be used within the OCCURS...DEPENDING ON construct.

```
05 asu-client-data PIC X(32).

05 asu-data-byte-count PIC 9(4) COMP.
05 asu-client-data PIC X(32).

01 add-subject-rec-rsps.
03 add-subject-rsp OCCURS 0 TO 1 TIMES DEPENDING ON add-subject-count.

* COPY rsp-air-add-item-rec-rsp OF "GLNK".

* Definition ADD-ITEM-REC-RSP created on 08/31/84 at 16:45

3< 05 rsp-air-add-item-rec-rsp.

3< 10 rsp-air-HDR.
3< 15 rsp-air-SELF-IDENT PIC AA.
3< 15 rsp-air-UOW-CODE PIC 9(4) COMP.
3< 10 rsp-air-RETN-CODE PIC S9(4) COMP.
3< 88 rsp-air-OK VALUE 0.
3< 88 rsp-air-E-BAD-TRANSACTION VALUE 4010.
3< 88 rsp-air-E-ITEM-NOT-FOUND VALUE 4035.
3< 88 rsp-air-E-ITEM-UNALTERABLE VALUE 4041.
3< 88 rsp-air-E-REC-ALREADY-EXISTS VALUE 4049.
3< 88 rsp-air-E-DATA-TOO-LONG VALUE 4085.
3< 88 rsp-air-E-INVALID-REC-TYPE VALUE 4046.
3< 88 rsp-air-E-INVALID-REC-SEQ-NUM VALUE 4098.
3< 10 rsp-air-RETN-CODE-DETAIL PIC S9(4) COMP.
3< 10 rsp-air-REC-SEQ-NUM PIC S9(4) COMP.
```

---

3.1.6 SCREEN COBOL Code for Client (Continued)
The following UOW, like all other special copies of the ADD-ITEM-REC UOW, must be "hand-coded" because the size of the field for client data varies with each use. Since the response to this UOW is completely fixed in length, however, the standard response definition supplied by Tandem can be used.

```cobol
01 add-recip-text-uow.
  05 art-hdr.
    10 art-self-ident PIC AA VALUE "UW".
    10 art-uow-code PIC 9(4) COMP VALUE 104.
  05 art-item-key.
    10 art-item-id.
      15 art-dummy PIC X(12).
    10 art-rec-type PIC S9(4) COMP VALUE 340.
    10 art-rec-seq-num PIC 59(4) COMP VALUE 1.
  05 art-data-byte-count PIC 9(4) COMP VALUE 70.
  05 art-client-data PIC X(70).

01 add-recip-text-rsp.
  * COPY rsp-air-add-ftem-rec-rsp OF "GCOB".
  Definition ADD-ITEM-REC-RSP created on 08/31/84 at 16:45
  05 rsp-air-ADD-ITEM-REC-RSP.
    10 rsp-air-HDR.
      15 rsp-air-SELF-IDENT PIC AA VALUE "UW".
      15 rsp-air-UOW-CODE PIC 9(4) COMP VALUE 104.
    10 rsp-air-RETN-CODE.
      88 rsp-air-OK.
      88 rsp-air-E-BAD-TRANSACTION.
      88 rsp-air-E-ITEM-NOT-FOUND.
      88 rsp-air-E-ITEM-UNALTERABLE.
      88 rsp-air-E-REC-ALREADY-EXISTS.
      88 rsp-air-E-DATA-TOO-LONG.
      88 rsp-air-E-INVALID-REC-TYPE.
      88 rsp-air-E-INVALID-REC-SEQ-NUM.
    10 rsp-air-RETN-CODE-DETAIL.
      10 rsp-air-REC-SEQ-NUM.
```

---

Sample Client
/ 01 add-text-array-count PIC 9(4) COMP.
01 add-text-rec-uows.
  03 add-text-uow-array OCCURS 0 TO 5 TIMES DEPENDING ON add-text-array-count.
  05 atx-hdr.
   10 atx-self-ident PIC AA.
   10 atx-uow-code PIC 9(4) COMP.
   05 atx-item-key.
   10 atx-item-id.
   15 atx-dummy PIC X(12).
  10 atx-rec-type PIC S9(4) COMP.
  10 atx-rec-seq-num PIC S9(4) COMP.
  05 atx-data-byte-count PIC 9(4) COMP.
  05 atx-client-data PIC X(78).
01 add-text-rec-rsps.
  03 add-text-rsp-array OCCURS 0 TO 5 TIMES DEPENDING ON add-text-array-count.
  05 rsp-air-add-item-rec-rsp OF "GLNK".
* Definition ADD-ITEM-REC-RSP created on 08/31/84 at 16:45
05 rsp-air-ADD-ITEM-REC-RSP.
  10 rsp-air-HDR.
  15 rsp-air-SELF-IDENT PIC AA.
  15 rsp-air-UOW-CODE PIC 9(4) COMP.
  10 rsp-air-RETN-CODE PIC S9(4) COMP.
  88 rsp-air-OK VALUE 4010.
  88 rsp-air-E-BAD-TRANSACTION VALUE 4035.
  88 rsp-air-E-ITEM-UNALTERABLE VALUE 4041.
  88 rsp-air-E-REC-ALREADY-EXISTS VALUE 4045.
  88 rsp-air-E-INVALID-REC-TYPE VALUE 4046.
  88 rsp-air-E-INVALID-REC-SEQ-NUm VALUE 4058.
  10 rsp-air-RETN-CODE-DETAIL PIC S9(4) COMP.
  10 rsp-air-REC-SEQ-NUm PIC S9(4) COMP.
Figure 7-5. SCREEN COBOL Code for Client (Continued)
Sample Client

Figure 7-5. SCREEN COBOL Code for Client (Continued)
01 create-item-uow.
   * COPY uow-cri-create-item-uow OF "GCOB".
2< 786  * Definition CREATE-ITEM-UOW created on 08/31/84 at 16:48
2< 787 05 uow-cri-CREATE-ITEM-UOW.
2< 788 10 uow-cri-HDR.
2< 789 15 uow-cri-SELF-IDENT PIC AA
2< 790 VALUE "UM".
2< 791 15 uow-cri-UOW-CODE PIC 9(4) COMP value 103.
2< 792 10 uow-cri-ITEM-TYPE PIC 9(4) COMP.
2< 793 10 uow-cri-IS-PKG-HDR PIC A.
2< 794 10 uow-cri-RESERVED-1 PIC A VALUE N".
2< 795 10 uow-cri-RESERVED-2 PIC A
2< 796 10 uow-cri-RESERVED-3 PIC A.

01 create-item-rsp.
   * COPY rsp-cri-create-item-rsp OF "GCOB".
2< 797  * Definition CREATE-ITEM-RSP created on 08/31/84 at 16:48
2< 798 05 rsp-cri-CREATE-ITEM-RSP.
2< 799 10 rsp-cri-HDR.
2< 800 15 rsp-cri-SELF-IDENT PIC AA
2< 801 VALUE "UM".
2< 802 15 rsp-cri-UOW-CODE PIC 9(4) COMP.
2< 803 10 rsp-cri-RETN-CODE PIC 9(4) COMP.
2< 804 88 rsp-cri-OK VALUE 0.
2< 805 88_rsp-cri-E-BAD-TRANSACTION VALUE 4010.
2< 806 88_rsp-cri-E-MUST-BE-YN VALUE 4051.
2< 807 88_rsp-cri-E-RESERVED-MUST-BE-N VALUE 4052.
2< 808 88_rsp-cri-E-INVALID-ITEM-TYPE VALUE 4056.
2< 809 10 rsp-cri-RETN-CODE-DETAIL PIC 9(4) COMP.
2< 810 10 rsp-cri-ITEM-ID.
2< 811 15 rsp-cri-DUMMY PIC X(12).

*********
01 submit-pkg-uow.
  05 uow-sp-SUBMIT-PKG-UOW.
  10 uow-sp-HDR.
  15 uow-sp-SELF-IDENT PIC AA.
  15 uow-sp-UOW-CODE VALUE "UW".
  10 uow-sp-ITEM-ID.
  15 uow-sp-DUMMY PIC X(12).

01 submit-pkg-rsp.
  05 rsp-sp-SUBMIT-PKG-RSP.
  10 rsp-sp-HDR.
  15 rsp-sp-SELF-IDENT PIC AA.
  15 rsp-sp-UOW-CODE VALUE "UW".
  10 rsp-sp-RETN-CODE.
  10 rsp-sp-RETN-CODE-DETAIL.

88 rsp-sp-OK VALUE 0.
88 rsp-sp-W-PRIORITY-REDUCED VALUE -4079.
88 rsp-sp-W-TIME-WINDOW-EXTENDED VALUE -4075.
88 rsp-sp-E-LIFESPAN-TOO-LONG VALUE 4077.
88 rsp-sp-E-BAD-TRANSACTION VALUE 4010.
88 rsp-sp-E-NO-RECIPS VALUE 4082.
88 rsp-sp-E-ITEM-NOT-FOUND VALUE 4035.
88 rsp-sp-E-ITEM-NOT-PKG-HDR VALUE 4042.
88 rsp-sp-E-NOT-CREATED-BY-YOU VALUE 4083.
88 rsp-sp-E-ITEM-UNALTERABLE VALUE 4041.
88 rsp-sp-E-TSCHED-UNAVAIL VALUE 4045.
88 rsp-sp-E-ITEM-TOD-COMPLEX VALUE 4036.
COPY uow-scn-SCAN-FOLDER-UOW OF "GCOB".

* Definition SCAN-FOLDER-UOW created on 08/31/84 at 16:53

05 uow-scn-SCAN-FOLDER-UOW.
200
15 uow-scn-HDR.
15 uow-scn-SELF-IDENT PIC AA VALUE "UW".
15 uow-scn-ITEM-ID.
10 uow-scn-ITEM-TYPE PIC 9(4) COMP.
15 uow-scn-ITEM-TYPE PIC 9(4) COMP.
10 uow-scn-DUMMY PIC X(12).
10 uow-scn-RESERVED-1 PIC A VALUE "N".
15 uow-scn-ITEM-TYPE PIC 9(4) COMP.
88 TANDEM-ASSIGNED values 100 thru 999.
88 uow-scn-ORIGINAL-PACKAGE value 109.
88 uow-scn-REPLY-PACKAGE value 110.
88 uow-scn-FORWARD-PACKAGE value 111.
88 uow-scn-UNFORMATTED-TEXT-ITEM value 120.
88 uow-scn-TTEXT-ITEM value 121.
88 uow-scn-ITEM-TYPE value 122.
88 uow-scn-ITEM-TYPE value 123.
88 uow-scn-ITEM-TYPE value 124.
88 uow-scn-ITEM-TYPE value 125.
88 uow-scn-ITEM-TYPE value 126.
88 uow-scn-ITEM-TYPE value 127.
88 uow-scn-ITEM-TYPE value 128.
88 uow-scn-ITEM-TYPE value 129.
88 uow-scn-COMPONENT-ITEM value 130.

* * * VALUE OF NUM-REQUESTED DETERMINES SIZE OF REPLY UOW (range 1:200) * * *
10 uow-scn-ITEM-TYPE PIC 9(4) COMP.

The response to this SCAN-FOLDER UOW was "hand-coded" to allow precise control over the maximum number of items returned (5) in the ITEMS-RETURNED field, below.

01 scan-folder-rsp.
02 rsp-scn-hdr.
03 rsp-scn-self-ident PIC AA VALUE "UW".
03 rsp-scn-uow-code PIC 9(4) COMP.
02 rsp-scn-retm-code PIC 9(4) COMP.
88 rsp-scn-OK VALUE 0.
88 rsp-scn-W-EOF VALUE -4001.
88 rsp-scn-W-TOO-MANY-REQUESTED VALUE -4091.
88 rsp-scn-E-INVALID-NUM-REQUESTED VALUE 4092.
88 rsp-scn-E-MUST-BE-YN VALUE 4051.
88 rsp-scn-E-RESERVED-MUST-BE-N VALUE 4052.
88 rsp-scn-E-FLD-ERROR-BAD-NAME VALUE 5677.
88 rsp-scn-E-FLD-BAD-BAD-NAME VALUE 5678.
88 rsp-scn-E-FLD-FLD-BAD-TYPE VALUE 5679.
88 rsp-scn-E-FLD-FLD-NAME VALUE 5680.
88 rsp-scn-E-FLD-FLD-ERROR-NAME VALUE 5681.
88 rsp-scn-E-FLD-FLD-ERROR-NAME VALUE 5682.
88 rsp-scn-E-FLD-FLD-ERROR-NAME VALUE 5683.
88 rsp-scn-E-FLD-FLD-ERROR-NAME VALUE 5684.
88 rsp-scn-E-FLD-FLD-ERROR-NAME VALUE 5685.
88 rsp-scn-E-FLD-FLD-ERROR-NAME VALUE 5686.
88 rsp-scn-E-FLD-FLD-ERROR-NAME VALUE 5687.
88 rsp-scn-E-FLD-FLD-ERROR-NAME VALUE 5688.
PAGE 16 $MERC.B9110.SMPCLNT TANDEM / SCREEN COBOL

232  88  rsp-scn-E-FLD-NO-PARENT  VALUE 5683.
233   02  rsp-scn-retn-code-detail PIC S9(4) COMP.
234   02  rsp-scn-num-returned  PIC 9(4) COMP.
235   02  rsp-scn-items-returned OCCURS 0 TO 5 TIMES
236       DEPENDING ON rsp-scn-num-returned OF
237               scan-folder-rsp.
238   03  rsp-scn-dummy  PIC X(12).
239       ********
01 get-item-descr-uow.
   * COPY uow-gid-get-item-descr-uow OF "GC0B".
2< 1007   * Definition GET-ITEM-DESCR-UOW created on 08/31/84 at 16:49
2< 1008   05 uow-gid-GET-ITEM-DESCR-UOW.
2< 1009   10 uow-gid-HDR.
2< 1010   15 uow-gid-SELF-IDENT PIC AA.
2< 1011   15 uow-gid-UID PIC 9(4) COMP value 122.
2< 1012   15 uow-gid-DUMMY PIC X(12).
2< 1013
2< 1014
2< 1015
2< 1016
2< 1017
2< 1018
2< 1019
2< 1020
2< 1021
2< 1022
2< 1023
2< 1024
2< 1025
2< 1026
2< 1027
2< 1028
2< 1029
2< 1030
2< 1031
2< 1032
2< 1033
2< 1034
2< 1035
2< 1036
2< 1037
2< 1038
2< 1039
2< 1040
2< 1041
2< 1042
2< 1043
2< 1044
2< 1045
2< 1046
2< 1047
2< 1048
2< 1049
2< 1050
2< 1051
2< 1052
2< 1053
2< 1054
2< 1055
2< 1056
2< 1057
2< 1058

01 get-item-descr-rsp.
   * COPY rsp-gid-get-item-descr-rsp OF "GC0B".
2< 1016   * Definition GET-ITEM-DESCR-RSP created on 08/31/84 at 16:49
2< 1017   05 rsp-gid-GET-ITEM-DESCR-RSP.
2< 1018   10 rsp-gid-HDR.
2< 1019   15 rsp-gid-SELF-IDENT PIC AA.
2< 1020   15 rsp-gid-UID PIC 9(4) COMP.
2< 1021   10 rsp-gid-RETN-CODE PIC S9(4) COMP.
2< 1022   88 rsp-gid-OK VALUE 0.
2< 1023   88 rsp-gid-E-ITEM-NOT-FOUND VALUE 4035.
2< 1024   10 rsp-gid-RETN-CODE-DETAIL PIC S9(4) COMP.
2< 1025   10 rsp-gid-UPDATE-CONTROL PIC S9(4) COMP.
2< 1026   10 rsp-gid-ITEM-TYPE PIC 9(4) COMP.
2< 1027   88 TANDEM-ASSIGNED values 100 thru 999.
2< 1028   88 rsp-gid-ORIGINAL-PACKAGE value 109.
2< 1029   88 rsp-gid-REPLY-PACKAGE value 110.
2< 1030   88 rsp-gid-FORWARD-PACKAGE value 111.
2< 1031   88 rsp-gid-UNIFORMATED-TEXT-ITEM value 120.
2< 1032   88 rsp-gid-TEXT-ITEM value 121.
2< 1033   88 rsp-gid-DATA-ITEM value 130.
2< 1034   88 rsp-gid-COMPOSITE-ITEM value 355.
2< 1035   88 rsp-gid-FACSIMILE-ITEM value 358.
2< 1036   10 rsp-gid-PARENT-COUNT PIC S9(4) COMP.
2< 1037   10 rsp-gid-COMPNT-COUNT PIC 9(4) COMP.
2< 1038   10 rsp-gid-ITEM-DESCR-FLAGS.
2< 1039   15 rsp-gid-IS-PKG-HDR PIC A.
2< 1040   15 rsp-gid-IS-PKG-HDR-REDEFINES rsp-gid-IS-PKG-HDR PIC X.
2< 1041   15 rsp-gid-UNALTERABLE PIC A.
2< 1042   15 rsp-gid-SUBMITTED PIC A.
2< 1043   15 rsp-gid-CANCELED PIC A.
2< 1044   15 rsp-gid-CLONED PIC A.
2< 1045   15 rsp-gid-REL-DATE-EARLIEST PIC A.
2< 1046   15 rsp-gid-REL-DATE-LATEST PIC A.
2< 1047   15 rsp-gid-REL-DATE-EXPIRATION PIC A.
2< 1048   15 rsp-gid-RESERVED-8 PIC A.
2< 1049   15 rsp-gid-RESERVED-9 PIC A.
2< 1050   15 rsp-gid-RESERVED-10 PIC A.
2< 1051   15 rsp-gid-RESERVED-11 PIC A.
2< 1052   15 rsp-gid-RESERVED-12 PIC A.
2< 1053   15 rsp-gid-RESERVED-13 PIC A.
The following fields are always returned:

- If the item is NOT a package header, they will contain binary zeros.

```
2< 1059 15 rsp-gid-RESERVED-13 PIC A
2< 1060 15 rsp-gid-RESERVED-14 VALUE "N" PIC A
2< 1061 15 rsp-gid-RESERVED-15 VALUE "N" PIC A
2< 1062 15 rsp-gid-RESERVED-16 PIC A
2< 1063 15 rsp-gid-RESERVED-17 PIC A
2< 1064 15 rsp-gid-RESERVED-18 PIC A
2< 1065 10 rsp-gid-CREATION-DATE PIC 9(4).
2< 1066 15 rsp-gid-YEAR PIC 9(2).
2< 1067 15 rsp-gid-MONTH PIC 9(2).
2< 1068 15 rsp-gid-DAY-OF-MONTH PIC 9(2).
2< 1069 15 rsp-gid-HOUR PIC 9(2).
2< 1070 15 rsp-gid-MINUTE PIC 9(2).
2< 1071 15 rsp-gid-SECOND PIC 9(2).
2< 1072 10 rsp-gid-CREATOR-NAME PIC X(120).
2< 1073 /* The following fields are always returned.
2< 1074 /* If the item is NOT a package header, they will contain binary zeros.
2< 1075 */
2< 1076 10 rsp-gid-SUBMITTED-DATE PIC 9(4).
2< 1077 15 rsp-gid-YEAR PIC 9(2).
2< 1078 15 rsp-gid-MONTH PIC 9(2).
2< 1079 15 rsp-gid-DAY-OF-MONTH PIC 9(2).
2< 1080 15 rsp-gid-HOUR PIC 9(2).
2< 1081 15 rsp-gid-MINUTE PIC 9(2).
2< 1082 15 rsp-gid-SECOND PIC 9(2).
2< 1083 10 rsp-gid-EARLIEST-DELIV-DATE.
2< 1084 10 rsp-gid-DATE-TIME.
2< 1085 15 rsp-gid-DATE-TIME.
2< 1086 20 rsp-gid-YEAR PIC 9(4).
2< 1087 20 rsp-gid-MONTH PIC 9(2).
2< 1088 20 rsp-gid-DAY-OF-MONTH PIC 9(2).
2< 1089 20 rsp-gid-HOUR PIC 9(2).
2< 1090 20 rsp-gid-MINUTE PIC 9(2).
2< 1091 20 rsp-gid-SECOND PIC 9(2).
2< 1092 15 rsp-gid-Delta-TIME REDEFINES rsp-gid-DATE-TIME.
2< 1093 20 rsp-gid-QUANTITY IC 9(4) COMP.
2< 1094 20 rsp-gid-QUANTITY PIC 9(4) COMP.
2< 1095 20 FILLER PIC X.
2< 1096 20 FILLER PIC X.
2< 1097 10 rsp-gid-LATEST-DELIV-DATE.
2< 1098 15 rsp-gid-DATE-TIME.
2< 1099 20 rsp-gid-YEAR PIC 9(4).
2< 1100 20 rsp-gid-MONTH PIC 9(2).
2< 1101 20 rsp-gid-DAY-OF-MONTH PIC 9(2).
2< 1102 20 rsp-gid-HOUR PIC 9(2).
2< 1103 20 rsp-gid-MINUTE PIC 9(2).
2< 1104 20 rsp-gid-SECOND PIC 9(2).
2< 1105 15 rsp-gid-Delta-TIME REDEFINES rsp-gid-DATE-TIME.
2< 1106 20 rsp-gid-QUANTITY PIC 9(4) COMP.
2< 1107 20 rsp-gid-QUANTITY PIC 9(4) COMP.
2< 1108 20 FILLER PIC X.
2< 1109 20 FILLER PIC X.
2< 1110 10 rsp-gid-EXPIRATION-DATE.
2< 1111 15 rsp-gid-DATE-TIME.
2< 1112 20 rsp-gid-YEAR PIC 9(4).
2< 1113 20 rsp-gid-MONTH PIC 9(2).
2< 1114 20 rsp-gid-DAY-OF-MONTH PIC 9(2).
2< 1115 20 rsp-gid-HOUR PIC 9(2).
```
<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2&lt; 1116</td>
<td>20 rsp-gid-MINUTE PIC 9(2).</td>
</tr>
<tr>
<td>2&lt; 1117</td>
<td>20 rsp-gid-SECOND PIC 9(2).</td>
</tr>
<tr>
<td>2&lt; 1118</td>
<td>15 rsp-gid-DELTA-TIME REDEFINES rsp-gid-DATETIME.</td>
</tr>
<tr>
<td>2&lt; 1119</td>
<td>20 rsp-gid-QUANTITY PIC 9(4) COMP.</td>
</tr>
<tr>
<td>2&lt; 1120</td>
<td>20 rsp-gid-UNITS PIC A.</td>
</tr>
<tr>
<td>2&lt; 1121</td>
<td>20 FILLER PIC X(10).</td>
</tr>
<tr>
<td>2&lt; 1122</td>
<td>10 rsp-gid-SENDER-ZONE-OFFSET PIC S9(4) COMP.</td>
</tr>
<tr>
<td>2&lt; 1123</td>
<td>10 rsp-gid-PRIORITY PIC 999 COMP.</td>
</tr>
<tr>
<td>2&lt; 1124</td>
<td>10 rsp-gid-APPLIC-ID PIC 9(4) COMP.</td>
</tr>
<tr>
<td>2&lt; 1125</td>
<td>10 rsp-gid-AGENT-SELECTOR PIC 9(4) COMP.</td>
</tr>
<tr>
<td>2&lt; 1126</td>
<td>10 rsp-gid-DELIV-CONTROL-FLAGS.</td>
</tr>
<tr>
<td>2&lt; 1127</td>
<td>15 rsp-gid-CERTIFIED PIC A.</td>
</tr>
<tr>
<td>2&lt; 1128</td>
<td>15 rsp-gid-RESERVED-1 PIC A.</td>
</tr>
<tr>
<td>2&lt; 1129</td>
<td>15 rsp-gid-RESERVED-2 PIC A.</td>
</tr>
<tr>
<td>2&lt; 1130</td>
<td>15 rsp-gid-RESERVED-3 PIC A.</td>
</tr>
<tr>
<td>2&lt; 1131</td>
<td>15 rsp-gid-RESERVED-4 PIC A.</td>
</tr>
<tr>
<td>2&lt; 1132</td>
<td>15 rsp-gid-RESERVED-5 PIC A.</td>
</tr>
<tr>
<td>2&lt; 1133</td>
<td>15 rsp-gid-RESERVED-6 PIC A.</td>
</tr>
<tr>
<td>2&lt; 1134</td>
<td>15 rsp-gid-RESERVED-7 PIC A.</td>
</tr>
<tr>
<td>2&lt; 1135</td>
<td>10 rsp-gid-INVALID-RECIPIENT PIC A.</td>
</tr>
<tr>
<td>2&lt; 1136</td>
<td>15 rsp-gid-INVALID-RECIPIENT PIC X.</td>
</tr>
<tr>
<td>2&lt; 1137</td>
<td>15 rsp-gid-INVALID-DLIST PIC A.</td>
</tr>
<tr>
<td>2&lt; 1138</td>
<td>10 rsp-gid-INVALID-DLIST PIC A.</td>
</tr>
<tr>
<td>2&lt; 1139</td>
<td>15 rsp-gid-INVALID-RECIPIENT PIC A.</td>
</tr>
<tr>
<td>2&lt; 1140</td>
<td>15 rsp-gid-INVALID-RECIPIENT PIC A.</td>
</tr>
<tr>
<td>2&lt; 1141</td>
<td>15 rsp-gid-INVALID-RECIPIENT PIC A.</td>
</tr>
<tr>
<td>2&lt; 1142</td>
<td>15 rsp-gid-INVALID-RECIPIENT PIC A.</td>
</tr>
<tr>
<td>2&lt; 1143</td>
<td>15 rsp-gid-INVALID-RECIPIENT PIC A.</td>
</tr>
<tr>
<td>2&lt; 1144</td>
<td>15 rsp-gid-INVALID-RECIPIENT PIC A.</td>
</tr>
<tr>
<td>2&lt; 1145</td>
<td>15 rsp-gid-INVALID-RECIPIENT PIC A.</td>
</tr>
<tr>
<td>2&lt; 1146</td>
<td>15 rsp-gid-INVALID-RECIPIENT PIC A.</td>
</tr>
<tr>
<td>2&lt; 1147</td>
<td>15 rsp-gid-INVALID-DLIST PIC A.</td>
</tr>
<tr>
<td>2&lt; 1148</td>
<td>15 rsp-gid-INVALID-DLIST PIC A.</td>
</tr>
<tr>
<td>2&lt; 1149</td>
<td>15 rsp-gid-INVALID-DLIST PIC A.</td>
</tr>
<tr>
<td>2&lt; 1150</td>
<td>15 rsp-gid-INVALID-DLIST PIC A.</td>
</tr>
<tr>
<td>2&lt; 1151</td>
<td>15 rsp-gid-INVALID-DLIST PIC A.</td>
</tr>
<tr>
<td>2&lt; 1152</td>
<td>15 rsp-gid-INVALID-DLIST PIC A.</td>
</tr>
<tr>
<td>2&lt; 1153</td>
<td>15 rsp-gid-INVALID-DLIST PIC A.</td>
</tr>
<tr>
<td>2&lt; 1154</td>
<td>15 rsp-gid-INVALID-DLIST PIC A.</td>
</tr>
<tr>
<td>2&lt; 1155</td>
<td>15 rsp-gid-INVALID-DLIST PIC A.</td>
</tr>
<tr>
<td>2&lt; 1156</td>
<td>15 rsp-gid-INVALID-DLIST PIC A.</td>
</tr>
<tr>
<td>2&lt; 1157</td>
<td>15 rsp-gid-INVALID-DLIST PIC A.</td>
</tr>
<tr>
<td>2&lt; 1158</td>
<td>10 FILLER PIC X(40).</td>
</tr>
</tbody>
</table>
/ 01 get-subject-rec-uow.
* Definition GET-ITEM-REC-UOW created on 08/31/84 at 16:50
 05 uow-gir-get-item-rec-uow.
 10 uow-gir-hdr.
 15 uow-gir-self-ident VALUE "UW".
 10 uow-gir-starting-key.
 15 uow-gir-item-id.
 20 uow-gir-dummy PIC X(12).
 15 uow-gir-rec-type PIC 9(4) COMP.
 15 uow-gir-rec-seq-num PIC 9(4) COMP.
 10 uow-gir-options.
 15 uow-gir-skip-exact PIC A.
 15 uow-gir-any-rec-type PIC A.
 15 uow-gir-any-seq-num PIC A.
 15 uow-gir-reserved-3 PIC A.
 10 uow-gir-num-requested.
 10 uow-gir-max-datasize.
 10 filler.
 01 get-subject-rec-rsp.
 02 gsu-hdr.
 03 gsu-self-ident PIC AA VALUE "UW".
 03 gsu-uow-code PIC 9(4) COMP.
 02 gsu-ret-n-code PIC S9(4) COMP.
 02 gsu-ret-code detail PIC S9(4) COMP.
 02 gsu-num-returned PIC 9(4) COMP.
 02 gsu-recs-returned OCCURS 0 TO 1 TIMES.
 03 gsu-rec-type PIC S9(4) COMP.
 03 gsu-rec-seq-num PIC S9(4) COMP.
 03 gsu-data-len PIC 9(4) COMP.
 03 gsu-data-string PIC X(32).
********

```
* * *
01 get-recip-text-uow.
  * COPY uow-gir-get-item-rec-uow OF "GCOB".
  05 uow-gir-GET-ITEM-REC-UOW.
  10 uow-gir-HDR.
    15 uow-gir-SELF-IDENT PIC AA VALUE "UW".
  10 uow-gir-STARTING-KEY.
    15 uow-gir-ITEM-ID.
  20 uow-gir-DUMMY PIC X(12).
  15 uow-gir-REC-TYPE PIC 9(4) COMP.
  15 uow-gir-SEQ-NUM PIC S9(4) COMP.
  10 uow-gir-OPTIONS.
    15 uow-gir-SKIP-EXACT PIC A.
    15 uow-gir-ANY-REC-TYPE PIC A.
    15 uow-gir-ANY-SEQ-NUM PIC A.
    15 uow-gir-RESERVED-3 PIC A.
  10 uow-gir-RESERVED-3 PIC A.
  10 uow-gir-NUM-REQUESTED VALUE OF NUM-REQUESTED AFFECTS SIZE OF REPLY UOW.
  10 uow-gir-MAX-DATASIZE VALUE OF MAX-DATASIZE AFFECTS SIZE OF REPLY UOW.
  10 uow-gir-PAD-CHAR.

01 get-recip-text-rsp.
  02 grt-hdr.
    03 grt-self-ident PIC AA VALUE "UW".
    02 grt-uow-code PIC 9(4) COMP.
  02 grt-retn-code PIC 59(4) COMP.
    88 grt-OK VALUE 0.
    88 grt-W-EOF VALUE -4001.
    88 grt-W-DATA-TRUNCATED VALUE -4089.
    88 grt-W-DATASIZE-ADJUSTED VALUE -4086.
    88 grt-W-TOO-MANY-REQUESTED VALUE 4091.
    88 grt-E-INVALID-NUM-RQSTD VALUE 4092.
    88 grt-E-INVALID-MAX-DATASIZE VALUE 4087.
    88 grt-E-MUST-BE-YN VALUE 4051.
    88 grt-E-RESERVED-MUST-BE-N VALUE 4052.
    88 grt-E-ITEM-NOT-FOUND VALUE 4035.
  02 grt-retn-code-detail PIC 59(4) COMP.
  02 grt-num-returned PIC 9(4) COMP.
    02 grt-recs-returned OCCURS 0 TO 1 TIMES DEPENDING ON grt-num-returned OF get-recip-text-rsp.
    03 grt-rec-type PIC 9(4) COMP.
    03 grt-rec-seq-num PIC 9(4) COMP.
    03 grt-data-len PIC 9(4) COMP.
    03 grt-data-string PIC X(70).
```

The response to this UOW was hand-coded to allow tailoring of the RECS-RETURNED and DATA-STRING fields to the application's requirements. Notice that in this UOW, the values entered in the NUM-REQUESTED and MAX-DATASIZE fields constrain the maximum number of occurrences for RECS-RETURNED and the length of the DATA-STRING field in the response returned by TISERV.
01 get-text-rec-uow.
   02 get-text-rec-rsp.
   03 gtx-hdr.
      03 gtx-self-ident PIC AA VALUE "UW".
      03 gtx-uow-code PIC 9(4) COMP.
   02 gtx-retn-code PIC 9(4) COMP.
      03 gtx-num-returned PIC 9(4) COMP.
      03 gtx-recs-returned OCCURS 0 TO 5 TIMES.
         03 gtx-rec-type PIC 9(4) COMP.
         03 gtx-rec-seq-num PIC 9(4) COMP.
         03 gtx-data-len PIC 9(4) COMP.
         03 gtx-data-string PIC X(78).
   02 get-text-rec-code-detail PIC 9(4) COMP.
   02 gtx-num-returned PIC 9(4) COMP.
   02 gtx-recs-returned OCCURS 0 TO 5 TIMES.
   03 gtx-rec-type PIC 9(4) COMP.
   03 gtx-rec-seq-num PIC 9(4) COMP.
   03 gtx-data-len PIC 9(4) COMP.
   03 gtx-data-string PIC X(78).

* Working storage area to contain item data records for display.
* These records are moved from gtx-data-string.
01 ws-data-rec-block.
  02 ws-data-recs OCCURS 5 TIMES.
  03 ws-data-rec-string PIC X(78).
*********
/ 01 ack-receipt-uow.
* COPY uow-ack-ack-receipt-uow OF "GCDB".
2< 318 Definition ACK-RECEIPT-UOW created on 08/31/84 at 16:45
2< 319 05 uow-ack-ACK-RECEIPT-UOW.
2< 320 10 uow-ack-HDR.
2< 321 15 uow-ack-SELX-IDENT PIC AA
2< 322 VALUE 'UW'.
2< 323 15 uow-ack-UOW-CODE PIC 9(4) COMP value 131.
2< 324 10 uow-ack-ITEM-ID.
2< 325 15 uow-ack-DUMMY PIC X(12).
2< 326 / 01 ack-receipt-rsp.
* COPY rsp-ack-ack-receipt-rsp OF "GCDB".
2< 327 Definition ACK-RECEIPT-RSP created on 08/31/84 at 16:45
2< 328 05 rsp-ack-ACK-RECEIPT-RSP.
2< 329 10 rsp-ack-HDR.
2< 330 15 rsp-ack-SELX-IDENT PIC AA
2< 331 VALUE 'UW'.
2< 332 15 rsp-ack-UOW-CODE PIC 9(4) COMP.
2< 333 10 rsp-ack-RETX-CODE PIC S9(4) COMP.
2< 334 88 rsp-ack-OK VALUE 0.
2< 335 88 rsp-ack-E-BAD-TRANSACTION VALUE 4010.
2< 336 88 rsp-ack-E-ITEM-NOT-FOUND VALUE 4035.
2< 337 88 rsp-ack-E-ITEM-NOT-PKG-HDR VALUE 4042.
2< 338 88 rsp-ack-E-PKG-NOT-SUBMITTED VALUE 4084.
2< 339 88 rsp-ack-E-PKG-NOT-RECEIVED VALUE 4080.
2< 340 88 rsp-ack-E-PKG-CANCELED VALUE 4094.
2< 341 88 rsp-ack-E-PKG-EXPIRED VALUE 4095.
2< 342 88 rsp-ack-E-TSCHED-UNAVAIL VALUE 4045.
2< 343 10 rsp-ack-RETX-CODE-DETAIL PIC S9(4) COMP.
/ 01 unsave-item-uow.
* COPY uow-usv-unsave-item-uow OF "GCOB".
2< 1896 * Definition UNSAVE-ITEM-UOW created on 08/31/84 at 16:55
2< 1897 05 uow-usv-UNSAVE-ITEM-UOW.
2< 1898 10 uow-usv-HDR.
2< 1899 15 uow-usv-SELF-IDENT PIC AA
2< 1900 15 uow-usv-UOW-CODE PIC 9(4) VALUE "UN".
2< 1901 10 uow-usv-ITEM-1D.
2< 1902 15 uow-usv-DUMMY PIC X(12).
2< 1903 10 uow-usv-FOLDER-NAME PIC X(80).
2< 1904
2< 1905
2< 1906
2< 1907
2< 1908
2< 1909
2< 1910
2< 1911
2< 1912
2< 1913
2< 1914
2< 1915
2< 1916
2< 1917
2< 1918
2< 1919
2< 1920
2< 1921
2< 1922
2< 1923
2< 1924
2< 1925
2< 1926
2< 1927
2< 1928
2< 1929
359 **********
**DEFINITIONS OF ERROR FIELDS AND VARIOUS FLAGS.**

** TRANSFER expects the time to be a PIC 9(6) format, however
** SCREEN COBOL returns a 9(8) format. The following lines are
** used to reformat from the SCREEN COBOL to the TRANSFER form.

```
05 ws-msg                    PIC X(25).
01 ws-error.
  05 error-name               PIC X(14).
  05 error-field-array.
    10 error-field OCCURS 6 TIMES.
    15 e-name                   PIC X(6).
    15 e-field                  PIC S9(4).
01 various-flags.
  05 term-status               PIC S9(4) VALUE ZERO.
  05 esub                      PIC 9(2) VALUE 1.
  05 key-pressed               PIC 9(2) VALUE 1.
  88 logon-request            VALUE 1.
  88 'function-request'       VALUE 3. 5.
  88 read-request             VALUE 3.
  88 discard-request          VALUE 4.
  88 compose-request          VALUE 5.
  88 post-request             VALUE 6.
  88 return-request           VALUE 7.
  88 logoff-request           VALUE 8.
  05 key-value                 PIC 9(2) VALUE 1.
  05 session-flag             PIC X(1) VALUE "N".
  05 session-active            VALUE "Y".
  05 item-flag                 PIC X(1) VALUE "N".
  05 item-active               VALUE "Y".
  05 exit-scan-flag            PIC X(1) VALUE "N".
  05 exit-scan                 VALUE "Y".
  05 first-scan-flag           PIC X(1) VALUE "Y".
  05 first-scan                VALUE "Y".
  05 air-index                 PIC 9(4) COMP.
  05 gtx-index                 PIC 9(4) COMP.
  05 scan-index                PIC 9(4) COMP.
```

```
SCREEN SECTION.

* DEFINITION OF BASE SCREEN. The logon, function menu, and 
* specific function screens, will be superimposed on this base 
* screen through overlays.

01 base-screen SIZE 24, 80.
   05 FILLER AT 1, 28
   REVERSE
   VALUE "E L E C - M A I L C L I E N T ".
   05 overlay-area AREA AT 3, 1 SIZE 20, 80.
   05 msg AT 23, 2
      PIC X(25)
      BLINK
      FROM ws-msg.
   05 error-line AT 24, 2
      PIC X(78)
      ADVISORY, REVERSE, DIM
      FROM ws-error.
431 /  
432 **************************************************************************  
433 * LOGON SCREEN OVERLAY                                             *  
434 **************************************************************************  
435 01 logon-screen OVERLAY SIZE 20, 80.  
436     05 FILLER AT 1, 31  
437     VALUE "-- LOGON SCREEN --"  
438 05 FILLER AT 5, 2  
439     VALUE "Please enter Name and Password:"
440 05 FILLER AT 8, 2  
441     VALUE "Name:"
442 05 name-in AT 8, 9  
443        PIC X(32)  
444        LENGTH MUST BE 1 THRU 32  
445        REVERSE, DIM  
446    TO uow-ss-corr-name .  
447 05 FILLER AT 10, 2  
448     VALUE "Password:"
449 05 password-in AT 10, 13  
450        PIC X(16)  
451        HIDDEN, REVERSE, DIM  
452    TO uow-ss-password .  
453 05 FILLER AT 20, 2  
454     VALUE "F1 = Logon    SF16 = Exit" .
455
Figure 7-5.
SCREEN COBOL Code for Client (Continued)

PAGE 29 $MERC.B9110.SMPCLNT TANDEM / SCREEN COBOL

456 01 select-function OVERLAY SIZE 20, 80 .
457   05 FILLER AT 1, 31
458   VALUE "-- FUNCTION MENU --" .
459   05 FILLER AT 4, 2
460   VALUE "Select the operation you wish to perform:" .
461   05 FILLER AT 6, 8
462   VALUE "F3 -- Read Mail" .
463   05 FILLER AT 7, 8
464   VALUE "F5 -- Send Mail" .
465   05 FILLER AT 8, 8
466   VALUE "SF16 -- Logoff and Exit" .
467   05 FILLER AT 9, 8
468   VALUE "F16 -- Logoff and Return" .
469   05 FILLER AT 10, 8
470   VALUE "--- FUNCTION MENU ---" .
I wrote.

c:

t)

I

en

-- MAIL POSTING SCREEN OVERLAY --

01 mail-posting OVERLAY SIZE 20, 80.
05 FILLER AT 1, 28
   VALUE "-- MAIL POSTING SCREEN --".
05 FILLER AT 4, 1
   VALUE "To:"
05 send-to-in AT 4, 11
   PIC X(32)
   LENGTH MUST BE 1 THRU 32
   REVERSE, DIM
   TO uow-ar-recip-name.
05 FILLER AT 6, 1
   VALUE "Subject:"
05 subject-in AT 6, 11
   PIC X(32)
   OCCURS ON 1 LINES DEPENDING ON add-subject-count
   REVERSE, DIM
   TO asu-client-data.
05 FILLER AT 9, 1
   VALUE "Text"
05 line-in AT 11, 2
   PIC X(78)
   OCCURS ON 5 LINES DEPENDING ON add-text-array-count
   REVERSE, DIM
   TO atx-client-data.
05 FILLER AT 20, 1
   VALUE "F6 -- Send"
05 FILLER AT 20, 15
   VALUE "F16 -- Quit and Return  SF16 -- Logoff".
**Figure 7-5. SCREEN COBOL Code for Client (Continued)**

```
507 /
508  *********************************************************************
509  * MAIL DISPLAY SCREEN OVERLAY                                      *
510  *********************************************************************
511  OF read-mail-screen OVERLAY SIZE 20, 80.
512      05 FILLER AT 1, 28
513      VALUE "-- MAIL DISPLAY SCREEN --".
514      05 FILLER AT 3, 1
515      VALUE "Sent:"
516      DIM.
517      05 month-out AT 3, 8
518      PIC 99
519      FROM rsp-gid-month OF rsp-gid-submitted-date.
520      05 FILLER AT 3, 11
521      VALUE "/".
522      05 day-out AT 3, 13
523      PIC 99
524      FROM rsp-gid-day-of-month OF rsp-gid-submitted-date.
525      05 FILLER AT 3, 16
526      VALUE "/".
527      05 year-out AT 3, 18
528      PIC 9(4)
529      FROM rsp-gid-year OF rsp-gid-submitted-date.
530      05 hour-out AT 3, 24
531      PIC 99
532      FROM rsp-gid-hour OF rsp-gid-submitted-date.
533      05 FILLER AT 3, 27
534      VALUE ":".
535      05 min-out AT 3, 29
536      PIC 99
537      FROM rsp-gid-minute OF rsp-gid-submitted-date.
538      05 FILLER AT 4, 1
539      VALUE "From:"
540      DIM.
541      05 from-out AT 4, 8
542      PIC X(32)
543      FROM rsp-gid-creator-name.
544      05 FILLER AT 5, 3
545      VALUE "To:"
546      DIM.
547      05 to-out AT 5, 8
548      PIC X(70)
549      FROM grt-data-string.
550      05 FILLER AT 6, 1
551      VALUE "Subj:".
552      DIM.
553      05 FILLER AT 6, 8
554      PIC X(32)
555      FROM gsu-data-string.
556      05 FILLER AT 8, 1
557      VALUE "Text:".
558      05 line-out AT 10, 2
559      PIC X(78)
560      OCCURS ON 5 LINES
561      REVERSE, DIM
562      FROM ws-data-recs.
```
Figure 7-5. SCREEN COBOL Code for Client (Continued)

```
564 05 FILLER AT 20, 1
      VALUE "F3 = Read Next".
566 05 FILLER AT 20, 19
567   VALUE "F4 = Discard".
568 05 FILLER AT 20, 36
569   VALUE "F16 = Return  SF16 = Logoff".
```
PROCEDURE DIVISION.

* MAIN ROUTINE.

DECLARATIVES.

logon-recovery SECTION.
USE FOR SCREEN RECOVERY.
DISPLAY base-screen.
END DECLARATIVES.

START-PROGRAM SECTION.
DISPLAY BASE base-screen.
sp-main-entry.
* key-value will be set to the correct value for branching
* to one of the following routines.
PERFORM ONE OF
logon-section,
function-section,
read-mail-section,
discard-mail-section,
compose-mail-section,
post-mail-section,
logoff-section,
logoff-section
DEPENDING ON key-value.

IF logoff-request
PERFORM logoff-section
GO TO sp-main-exit
ELSE
GO TO sp-main-entry.
sp-main-exit.
EXIT PROGRAM.
Figure 7-5. SCREEN COBOL Code for Client (continued)

614 /logon-section SECTION.
615
616
617 The logon screen is displayed and either function key 1 (F1) or
618 shifted function key 16 (SF16) will be accepted.
619
620 A F1 key is used to enter the logon, SF16 will exit, all other
621 keys are errors.
622
623 Following successful logon, key-value is set to 2 to force
624 branching to the select screen. A SF16 will invoke the logoff
625 function by setting key-value to 6.
626
627******************************************************************************
628
629 DISPLAY OVERLAY logon-screen AT overlay-area.
630
631 logon-display-data.
632
633 DISPLAY logon-screen.
634
635 logon-accept-data.
636
637 ACCEPT logon-screen
638 UNTIL logon-key, f2, read-mail, discard-mail,
639 compose-mail, post-mail, return-key
640 ESCAPE ON
641 logoff-key.
642
643 MOVE TERMINATION-STATUS TO key-pressed.
644
645 If logoff-request
646 GO TO logon-section-exit.
647
648 IF NOT logon-request
649 MOVE "FUNCTION KEY PRESSED IS NOT ALLOWED" TO ws-error
650 DISPLAY TEMP error-line of base-screen
651 GO TO logon-accept-data.
652
653 DISPLAY TEMP "LOGGING ON" in msg.
654
655******************************************************************************
656
657 (1) SETS UP THE IPC HEADER IN PREPARATION FOR STARTING A
658 SESSION. A client must establish a session as the first
659 act performed. The SESSION-ID field must be set to binary
660 zero for TRANSFER to grant a session.
661
662******************************************************************************
663
664 MOVE -3 TO ih-request-code.
665 MOVE LOW-VALUES TO ih-session-id.
666 MOVE 1 TO ih-uows-to-process.
667
668 To trace errors, set this field to "Y". For
669 normal operation, set to "N".
670
MOVE "N" TO ih-log-this-ipc.

MOVE "START-SESSION" TO error-name.
MOVE ZERO TO uow-ss-applic-id.
MOVE 19 TO uow-ss-century.
ACCEPT uow-ss-accept-date FROM DATE.
ACCEPT accept-time FROM TIME.
MOVE accept-time1 TO uow-ss-accept-time.

BEGIN-TRANSACTION.

* A -3 in the IPC header forces TRANSFER to process all UOWs passed. However, if the ipc header is not valid, the reply will contain only the ipc-hdr.

*********************************************************************
*(2) SENDS THE IPC TO START A SESSION (within the framework of a TMF transaction).
*********************************************************************

SEND ipc-hdr, start-session-uow TO "TISERV"
REPLY CODE 0, 1, 2 YIELDS ipc-hdr, start-session-rsp
CODE 3 YIELDS ipc-hdr
ON ERROR MOVE TERMINATION-STATUS TO term-status
PERFORM error-section
ABORT-TRANSACTION
GO TO logon-accept-data.

IF ih-pw-reply-code > 1
PERFORM error-section
ABORT-TRANSACTION
GO TO logon-accept-data.

END-TRANSACTION.

RESER TEMP msg.
MOVE "Y" TO session-flag.
MOVE 2 TO key-value.

*********************************************************************
*(3) EXITS THE LOGON SECTION.
*********************************************************************

logon-section-exit.
EXIT.
/ function-section SECTION.

*********************************************************************
* This function displays the function selection menu and accepts      *
* a function key. In this example, not all function keys are        *
* tested -- those which are not included in the accept list will    *
* not be recognized                                                  *
*********************************************************************

RESET TEMP msg.
display-function.

DISPLAY OVERLAY select-function AT overlay-area.
accept-function.

ACCEPT
ESCAPE ON
logon-key, f2, read-mail, discard-mail,
compose-mail, post-mail
return-key, logoff-key.

MOVE TERMINATION-STATUS TO key-pressed.

IF return-request OR logoff-request
MOVE 8 TO key-value
GO TO function-section-exit.

IF NOT function-request
MOVE "FUNCTION KEY PRESSED IS NOT ALLOWED" TO ws-error
DISPLAY TEMP error-line OF base-screen
GO TO accept-function.

MOVE key-pressed TO key-value.

function-section-exit.
EXIT.
read-mail-section SECTION.

DISPLAY OVERLAY read-mail-screen AT overlay-area.

MOVE 115 TO uow-gir-rec-type OF get-subject-rec-uow.
MOVE 1 TO uow-gir-rec-seq-num OF get-subject-rec-uow.
MOVE ALL 'N' TO uow-gir-options OF get-subject-rec-uow.
MOVE 1 TO uow-gir-num-requested OF get-subject-rec-uow.
MOVE 32 TO uow-gir-max-datasize OF get-subject-rec-uow.
MOVE SPACE TO uow-gir-pad-char OF get-subject-rec-uow.

MOVE 340 TO uow-gir-rec-type OF get-recip-text-uow.
MOVE 1 TO uow-gir-rec-seq-num OF get-recip-text-uow.
MOVE ALL 'N' TO uow-gir-options OF get-recip-text-uow.
MOVE 1 TO uow-gir-num-requested OF get-recip-text-uow.
MOVE 70 TO uow-gir-max-datasize OF get-recip-text-uow.
MOVE SPACE TO uow-gir-pad-char OF get-recip-text-uow.

MOVE 120 TO uow-gir-rec-type OF get-text-rec-uow.
MOVE 1 TO uow-gir-rec-seq-num OF get-text-rec-uow.
MOVE ALL 'N' TO uow-gir-options OF get-text-rec-uow.
MOVE 5 TO uow-gir-num-requested OF get-text-rec-uow.
MOVE 78 TO uow-gir-max-datasize OF get-text-rec-uow.
MOVE SPACE TO uow-gir-pad-char OF get-text-rec-uow.

MOVE LOW-VALUES TO uow-scn-item-id.
MOVE "INBOX" TO uow-scn-folder-name.
MOVE "N" TO uow-scn-filter-by-item-type.
MOVE 5 TO uow-scn-num-requested.

MOVE 2 TO key-value.
MOVE 'N' TO exit-scan-flag.
MOVE 'Y' TO first-scan-flag.
MOVE 0 TO rsp-scn-retn-code.

PERFORM scan-mail-section
UNTIL rsp-scn-w-eof OR exit-scan.

IF rsp-scn-num-returned = 0 AND first-scan
DISPLAY TEMP "NO ITEMS FOUND" IN error-line OF base-screen
ELSE
DISPLAY TEMP "NO MORE ITEMS" IN error-line OF base-screen.
**Figure 7-5. SCREEN COBOL Code for Client (Continued)**

```cobol
810  / scan-mail-section SECTION.
811  *
812  * Builds the UOW to scan the "INBOX" to find items associated *
813  * with this correspondent.
814  *
815  *
816  *
817  *
818  DISPLAY TEMP "SCANNING FILE" IN msg.
819  MOVE "SCAN" TO error-name.
820  MOVE -3 TO ih-request-code.
821  MOVE 1 TO ih-nums-to-process.
822  *
823  *
824  * SENDS THE IPC REQUESTING THE FOLDER SCAN. Because this *
825  * operation does not affect the data base, no TMF transaction *
826  * is necessary.
827  *
828  *
829  SEND IPC-HDR, scan-folder-uow
830  TO "TISERV"
831  REPLY CODE 0, 1, 2 YIELDS ipc-hdr, scan-folder-rsp,
832  CODE 3 YIELDS ipc-hdr.
833  ON ERROR MOVE TERMINATION-STATUS TO term-status
834  PERFORM error-section
835  GO TO scan-mail-section-exit.
836  IF ih-pw-reply-code > 1
837  PERFORM error-section
838  MOVE "Y" TO exit-scan-flag
839  GO TO scan-mail-section-exit.
840  IF rsp-scn-num-returned = 0
841  MOVE "Y" TO exit-scan-flag
842  GO TO scan-mail-section-exit.
843  RESET TEMP msg.
844  PERFORM get-mail-section
845  VARYING scan-index FROM 1 BY 1
846  UNTIL scan-index > rsp-scn-num-returned OR exit-scan.
847  scan-mail-section-exit.
848  EXIT.
```
Figure 7-5. SCREEN COBOL Code for Client (continued)

855  / get-mail-section SECTION.
856
857  858  *************************************************************************
859  * Builds the UOW to get the data and display one package from the *
860  * list.  *
861  *************************************************************************
862
863  BEGIN-TRANSACTION.
864  865  MOVE "N" TO item-flag.
866  867  MOVE rsp-scn-items-returned( scan-index )
868  869  TO uow-gid-item-id,
870  871  uow-gir-item-id OF get-subject-rec-uow,
872  873  uow-gir-item-id OF get-recip-text-uow,
874  875  uow-gir-item-id OF get-text-rec-uow,
876  877  uow-ack-item-id.
878  879  MOVE "GET ITEMS" TO error-name.
880  881  MOVE -3 TO ih-request-code.
882  883  MOVE 5 TO ih-uows-to-process.
884  885  *************************************************************************
886  * SENDS THE IPC TO GET THE DATA AND DISPLAY THE PACKAGE. Because *
887  * this request includes the ACK-RECEIPT UOW along with the other *
888  * data-retrieval UOWs, a TMF transaction is required. *
889  *************************************************************************
890  891  SEND ipc-hdr,
892  893  get-item-descr-uow,
894  895  get-subject-rec-uow,
896  897  get-recip-text-uow,
898  899  get-text-rec-uow,
900  901  ack-receipt-uow
902  903  TO "TISERV"
904  905  REPLY CODE 0, 1, 2 YIELDS ipc-hdr,
906  907  get-item-descr-rsp,
908  909  get-subject-rec-rsp,
910  911  get-recip-text-rsp,
912  913  get-text-rec-rsp,
914  915  ack-receipt-rsp
916  917  CODE 3 YIELDS ipc-hdr
918  919  ON ERROR MOVE TERMINATION-STATUS TO term-status
920  921  PERFORM error-section
922  923  ABORT-TRANSACTION
924  925  GO TO accept-mail-data.
926  927  928  IF ih-pw-reply-code > 1
929  930  PERFORM error-section
931  932  ABORT-TRANSACTION
933  934  GO TO accept-mail-data.
935  936  END-TRANSACTION.
937  938  IF gsu-num-returned < 1
939  940  MOVE 1 TO gsu-num-returned
941  942  MOVE SPACES TO gsu-data-string( 1 ).
IF grt-num-returned < 1
  MOVE 1 TO grt-num-returned
  MOVE SPACES TO grt-data-string(1).
MOVE SPACES TO ws-data-rec-block.
PERFORM move-gtx-data-string
  VARYING gtx-index FROM 1 BY 1
  UNTIL gtx-index > gtx-num-returned.
DISPLAY read-mail-screen.
IF rsp-scn-w_eof AND scan-index = rsp-scn-num-returned
  DISPLAY TEMP "LAST ITEM" IN error-line OF base-screen.
MOVE "Y" TO item-flag.
accept-mail-data.
ACCEPT
  ESCAPE ON
    logon-key, f2, read-mail, discard-mail,
    compose-mail, post-mail
  return-key, logoff-key.
MOVE TERMINATION-STATUS TO key-pressed.
IF return-request
  MOVE 2 TO key-value
  MOVE "Y" TO exit-scan-flag
  GO TO get-mail-section-exit.
IF logoff-request
  MOVE 8 TO key-value
  MOVE "Y" TO exit-scan-flag
  GO TO get-mail-section-exit.
IF discard-request AND item-active
  PERFORM discard-mail-section
  GO TO get-mail-section-exit.
IF NOT read-request
  MOVE "FUNCTION KEY PRESSED IS NOT ALLOWED" TO ws-error
  DISPLAY TEMP error-line OF base-screen
  GO TO accept-mail-data.
IF item-active
  MOVE rsp-scn-items-returned( scan-index ) TO uow-scn-item-id.
GO TO get-mail-section-exit.
move-gtx-data-string.
MOVE gtx-data-string( gtx-index ) TO ws-data-rec-string( gtx-index ).
get-mail-section-exit.
EXIT.
BEGIN-TRANSACTION.

MOVE "DISCARD" TO error-name.
MOVE 1 TO ih-uows-to-process.
MOVE -3 TO ih-request-code.
MOVE "INBOX" TO uow-usv-folder-name.
MOVE rsp-scn-items-returned( scan-index ) TO uow-usv-item-id.
DISPLAY TEMP "DISCARDING" IN msg.

PEND TEMP "DISCARDING" IN msg.

SEND ipc-hdr, unsave-item-uow TO "TISERV" REPLY CODE 0, 1, 2 YIELDS ipc-hdr, unsave-item-rsp
CODE 3, YIELDS ipc-hdr,
ON ERROR MOVE TERMINATION-STATUS TO term-status
PERFORM error-section
ABORT-TRANSACTION
GO TO discard-mail-section-exit.

IF ih-pw-reply-code > 1
PERFORM error-section
ABORT-TRANSACTION
GO TO discard-mail-section-exit.
RESET TEMP msg.
END-TRANSACTION.

MOVE "PREVIOUS ITEM DISCARDED" TO WS-MSG
DISPLAY TEMP msg.
discard-mail-section-exit.
EXIT.
compose-mail-section SECTION.

******************************************************************************
* Allows correspondent to compose mail for subsequent transmittal. *
******************************************************************************

DISPLAY OVERLAY mail-posting AT overlay-area.
DISPLAY mail-posting.
accept-mail-posting.

ACCEPT mail-posting
UNTIL logon-key, f2, read-mail, discard-mail,
compose-mail, post-mail
ESCAPE ON return-key, logoff-key.

MOVE TERMINATION-STATUS TO key-pressed.

IF return-request
MOVE 2 TO key-value
GO TO compose-mail-section-exit.

IF logoff-request
MOVE 8 TO key-value
GO TO compose-mail-section-exit.

IF NOT post-request
MOVE "FUNCTION KEY PRESSED IS NOT ALLOWED" TO ws-error
DISPLAY TEMP error-line OF base-screen
GO TO accept-mail-posting.

PERFORM post-mail-section.

compose-mail-section-exit.
EXIT.
post-mail-section SECTION.

BEGIN-TRANSACTION.

DISPLAY TEMP "POSTING" IN msg.

* Describe the package to create.
MOVE 109 TO uow-cri-item-type.
MOVE "Y" TO uow-cri-is-pkg-hdr.
MOVE "CREATE" TO error-name.
MOVE -3 TO ih-request-code.
MOVE 1 TO ih-uows-to-process.
SEND IPC-HDR, create-item-uow TO "TISERV"
REPLY CODE 0, 1, 2 YIELDS ipc-hdr, create-item-rsp
YIELDS ipc-hdr ON ERROR MOVE TERMINATION-STATUS TO term-status
PERFORM error-section
ABORT-TRANSACTION
GO TO post-mail-section-exit.

IF ih-pw-reply-code > 1
PERFORM error-section
ABORT-TRANSACTION
GO TO post-mail-section-exit.

* Setup the data in each uow to be sent
MOVE rsp-cri-item-id TO uow-ar-item-id.
MOVE 340 TO uow-ar-recip-type.
MOVE ALL "N" TO uow-ar-options.
MOVE "Y" TO uow-ar-use-depot-resol-flags.
MOVE "ADD ITEMS" TO error-name.
MOVE -3 TO ih-request-code.
COMPUTE ih-uows-to-process = (1 + add-text-array-count + add-subject-count).
* SENDS THE IPC TO ADD RECIPIENT, SUBJECT, AND TEXT RECORDS TO PACKAGE. *
* **********************************************
  SEND ipc-hdr,
    add-recip-uow,
    add-subject-rec-uows,
    add-text-rec-uows
  TO "TISERV"
  REPLY CODE 0, 1, 2 YIELDS ipc-hdr,
    add-recip-rsp,
    add-subject-rec-rsps,
    add-text-rec-rsps.
  CODE 3, YIELDS ipc-hdr,
  ON ERROR MOVE TERMINATION-STATUS TO term-status
  PERFORM error-section
  ABORT-TRANSACTION
  GO TO post-mail-section-exit.

  IF ih-pw-reply-code > 1
    PERFORM error-section
  ABORT-TRANSACTION
  GO TO post-mail-section-exit.
  MOVE rsp-cri-item-id TO art-item-id, uow-sp-item-id.
  MOVE rsp-ar-accepted-name TO art-client-data.
  MOVE -3 TO ih-request-code.
  MOVE 2 TO ih-uows-to-process.

* SENDS THE IPC TO POST THE PACKAGE. *
* **********************************************
  SEND ipc-hdr,
    add-recip-text-uow,
    submit-pkg-uow
  TO "TISERV"
  REPLY CODE 0, 1, 2 YIELDS ipc-hdr,
    add-recip-text-rsp,
    submit-pkg-rsp
  CODE 3, YIELDS ipc-hdr,
  ON ERROR MOVE TERMINATION-STATUS TO term-status
  PERFORM error-section
  ABORT-TRANSACTION
  GO TO post-mail-section-exit.

  IF ih-pw-reply-code > 1
    PERFORM error-section
  ABORT-TRANSACTION
  GO TO post-mail-section-exit.
  END-TRANSACTION.
MOVE "PACKAGE POSTED" TO ws-error
DISPLAY TEMP error-line OF base-screen.
RESET TEMP msg.
GO TO post-mail-section-exit.

init-uow-air.

* Setup the fields in each text add-item-rec uow.
MOVE "UW" TO atx-self-ident( air-index ).
MOVE 104 TO atx-uow-code( air-index ).
MOVE rsp-cri-item-id TO atx-item-id( air-index ).
MOVE 120 TO atx-rec-type( air-index ).
MOVE air-index TO atx-rec-seq-num( air-index ).
MOVE 78 TO atx-data-byte-count( air-index ).
post-mail-section-exit.
EXIT.
logoff-section SECTION.

*********************************************************************
Provides for termination of dialog between correspondent and TRANSFER.
*********************************************************************
IF NOT session-active
  GO TO logoff-section-exit.
  TURN TEMP BLINK IN msg
  DISPLAY TEMP "LOGGING OFF" IN msg.
BEGIN-TRANSACTION.
MOVE "LOGOFF" TO error-name.
MOVE -3 TO ih-request-code.
MOVE 1 TO ih-uows-to-process.
*********************************************************************
SENDS IPC TO END SESSION.
*********************************************************************
SEND ipc-hdr, end-session-uow TO "TISERV"
  TO "TISERV"
  ON ERROR MOVE TERMINATION-STATUS TO term-status
  PERFORM error-section
  ABORT-TRANSACTION.
MOVE 1 to key-value.
MOVE "N" TO session-flag.
RESET TEMP msg.
END-TRANSACTION.
DISPLAY TEMP "LOGOFF COMPLETED" IN error-line OF base-screen.
logoff-section-exit.
EXIT.
error-section SECTION.

*********************************************************************
* Only rudimentary error handling is provided in this example.      *
* If errors are found, refer to the TRANSFER programming guide.    *
*********************************************************************

RESET TEMP msg.

MOVE SPACES TO error-field-array.

MOVE 1 TO ESUB.

IF term-status NOT = 0
  MOVE "TM-ST" TO e-name (esub)
  MOVE term-status TO e-field (esub)
  ADD 1 TO esub.

IF ih-pw-reply-code NOT = 0
  MOVE "PWRP" TO e-name (esub)
  MOVE ih-pw-reply-code TO e-field (esub)
  ADD 1 TO esub.

IF ih-ipc-retm-code NOT = 0
  MOVE "PWRT" TO e-name (esub)
  MOVE ih-ipc-retm-code TO e-field (esub)
  ADD 1 TO esub.

IF error-name = "START-SESSION"
  IF NOT rsp-ss-ok
    MOVE "SSRT" TO e-name (esub)
    MOVE rsp-ss-retm-code TO e-field (esub)
    ADD 1 TO esub
    IF rsp-ss-retm-code-detail NOT = 0
      MOVE "SSRD" TO e-name (esub)
      MOVE rsp-ss-retm-code-detail TO e-field (esub)
      ADD 1 TO esub.

IF error-name = "CREATE"
  IF NOT rsp-cri-ok
    MOVE "CRI" TO e-name (esub)
    MOVE rsp-cri-retm-code TO e-field (esub)
    ADD 1 TO esub.

IF error-name = "ADD ITEMS"
  IF NOT rsp-ar-ok
    MOVE "A-R" TO e-name (esub)
    MOVE rsp-ar-retm-code TO e-field (esub)
    ADD 1 TO esub.

DISPLAY TEMP error-line.

error-section-exit.
EXIT.
PAGE 48 $MERC.B9110.SMPCLNT

OBJECT FILE NAME IS $MERC.BETHEXMP.EXAMP
PROGRAM NAME IS SAMPLE-CLIENT
PROGRAM VERSION IS 1
NO. ERRORS=0; NO. WARNINGS=0
CODE SIZE=2654
DATA SIZE=3284
NUMBER OF SOURCE LINES READ=1876
MAXIMUM SYMBOL TABLE SIZE=19936 WORDS
ELAPSED TIME - 0:07:50
SECTION 8
SAMPLE AGENT

This section presents an example of the operations performed by a very simple agent. In this example, the agent:

1. forwards each package received at the depot to another correspondent
2. removes the package from the depot's INBOX folder
3. acknowledges receipt of the package.

The correspondent to whom the packages are forwarded is specified when the agent is configured.

The SCREEN COBOL source code required for this agent appears in Figure 8-1. The sample agent uses the following UOWs:

- ACK-RECEIPT
- ADD-ITEM-REC
- ATTACH-COMPNT-A01
- CREATE-ITEM
- SUBMIT-PKG
- UNSAVE-ITEM
- ADD-RECIPI

Through 01 record level definitions, this code illustrates how definitions from the GCOB and GLNK files can be copied into a program during compilation. The GCOB file contains COBOL source code for commonly used TRANSFER elements, such as IPC header and UOW definitions. The GLNK file contains similar code for use in the Linkage Section of the program; in this file, however, the INITIAL-VALUE clauses have been removed to comply with the requirements of the Linkage Section.

Notice that the names of GLNK fields that appear in the USING clause in the first line of the PROCEDURE Division (Line 101) must match exactly the names appearing in the Linkage Section (Lines 95 and 98).
Sample Agent

For clarity, this example is coded in a very straightforward way and is explained by comments within the source program code. To reduce storage requirements, you could use the REDEFINES clause to redefine certain UOWs and their responses to occupy the same storage space; you would do this, however, only when you were certain that the redefined space was no longer needed for its original purpose.

Source code for the sample agent has been supplied by Tandem and is on your system. The file is named SMPAGNT. Unlike the sample client, instructions are not provided for configuring this agent. To have the sample agent installed in your TRANSFER system, see your system manager. The paragraph titled DESIGNING AND WRITING AN AGENT in Section 6 provides instructions. After the sample agent is installed, it can be configured for a depot with the default agent configuration screen described in the TRANSFER Delivery System Management and Administration Guide.
Figure 8-1. SCREEN COBOL Code for Agent

Sample Agent

```
IDENTIFICATION DIVISION.
PROGRAM-ID. SAMPLE-AGENT.
* "SAMPLE-AGENT" is the name you would specify on the screen titled,
* "Agent Selection".

AUTHOR. TANDEM
INSTALLATION.
DATE-WRITTEN. 84/09/17 - 17:23:50.

**********************************************************************
* This Sample Agent will do the following:
* (1) Forward all incoming packages to a person specified
* in the Agent Configuration (See the screen titled
* "Default Agent Configuration", the field titled,
* "Enter any data needed by the agent on the following
* line:").
* (2) Remove the package from the INBOX.
* (3) Acknowledge that the package was examined.
**********************************************************************
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. T16.
OBJECT-COMPUTER. T16.
```
DATA DIVISION.

WORKING-STORAGE SECTION.

***************************************************
* IPC MESSAGE DEFINITIONS

01 ipc-hdr-block.
*
COPY ih-ipc-hdr OF gcob".
*
Definition IPC-HDR created on 08/31/84 at 16:44
*
05 ih-IPC-HDR.
* O1 ih-IPC-HDR.
  10 ih-REQUEST-CODE PIC 99(4) COMP.
  10 ih-STOP-ON-WARNING Value is -1.
  10 ih-STOP-ON-ERR Value is -2.
  10 ih-DO-ALL-UDWS Value is -3.
  10 ih-PW-REPLV-CODE REDEFINES ih-ALL-UDWS(4) COMP.
  10 ih-ALL-UDWS-OK Value is 0.
  10 ih-UDWS-WITH-WARNING Value is 1.
  10 ih-UDWS-WITH-ERR Value is 2.
  10 ih-RQST-ERR Value is 3.
  10 FILLER PIC S9(4) Value is 0.
  10 ih-VERSION-CODE.
  15 ih-LETTER PIC A value "B".
  15 ih-REV-NUMBER PIC 99 value 00.
  10 ih-IPC-RETN-CODE PIC S9(4) COMP.
  10 ih-IPC-OK Value is 0.
  10 ih-INVALID-VERSION-CODE Value is 1.
  10 ih-INVALID-SESSION-ID Value is 2.
  10 ih-SERVICE-DENIED Value is 3.
  10 ih-INVALID-UDW-HDR Value is 4.
  10 ih-RQST-TOD-LONG Value is 5.
  10 ih-REPLY-TOD-LONG Value is 6.
  10 ih-RQST-TOD-SHORT Value is 7.
  10 ih-INVALID-REQUEST-CODE Value is 8.
  10 ih-INVALID-REQUEST-CODE-DETAIL PIC S9(4) Value 0.
  10 ih-E-BAD-TRANSACTION VALUE 4010.
  10 ih-E-ERR-PROFILE-FILE VALUE 4902.
  10 ih-E-ERR-SESSION-FILE VALUE 4904.
  10 ih-E-ERR-ITEMDESC-FILE VALUE 4906.
  10 ih-E-ERR-ITEMDATA-FILE VALUE 4908.
  10 ih-E-ERR-RECIPIENT-FILE VALUE 4910.
  10 ih-E-ERR-FOLDER-FILE VALUE 4912.
  10 ih-E-ERR-ENLIST-FILE VALUE 4914.
  10 ih-E-ERR-READY-FILE VALUE 4916.
  10 ih-E-ERR-SECTION-FILE VALUE 4918.
  10 ih-E-ERR-FILE VALUE 4920.
  10 ih-E-ERR-INV-FOLDER-FILE VALUE 4922.
  10 ih-E-ERR-QUEUE-FILE VALUE 4924.
  10 ih-E-IO-TIMEOUT VALUE 4990.
* For IPC-RETN-CODE = E-BAD-TRANSACTION or E-ERR-FILE-xxxx, this
  field contains the GUARDIAN error code that describes the problem.
* For IPC-RETN-CODE = E-IO-TIMEOUT, this field contains the file
  code of the file on which the timeout occurred.
* For other values of IPC-RETN-CODE, this field is undefined.
  10 ih-IPC-RETN-CODE-DETAIL PIC S9(4) COMP.
  10 ih-IPC-OK Value is 0.
PAGE 3 $MERC.B9110.SMPAGNT TANDEM / SCREEN COBOL

Figure 8-1. SCREEN COBOL Code for Agent (Continued)

***************
51 01 uow-ack-block.
52 * COPY uow-ack-ack-receipt-uow OF "gcob".
53
54 * Definition ACK-RECEIPT-UOW created on 08/31/84 at 16:45
55
56 2< 318 05 uow-ack-ACK-RECEIPT-UOW.
57 2< 319 10 uow-ack-HDR.
58 2< 320 15 uow-ack-SELF-IDENT PIC AA
59 2< 321 VALUE "UW".
60 2< 322 15 uow-ack-UOW-CODE PIC 9(4) COMP value 131.
61 2< 323 20 uow-ack-DUMMY PIC X(12).
62
***************
63 01 uow-air.
64 * COPY uow-air-add-item-rec-uow OF "gcob".
65
66 * Definition ADD-ITEM-REC-UOW created on 08/31/84 at 16:45
67
68 2< 345 05 uow-air-ADD-ITEM-REC-UOW.
69 2< 346 10 uow-air-HDR.
70 2< 347 15 uow-air-SELF-IDENT PIC AA
71 2< 348 VALUE "UW".
72 2< 349 15 uow-air-UOW-CODE PIC 9(4) COMP value 104.
73
***************
74 01 ih-SESSION-ID.
75 10 ih-DUMMY PIC X(18).
76 2< 186 10 ih-UDWS-TO-PROCESS PIC 9(4) COMP.
77 2< 187 10 ih-UDWS-RETURNED PIC 9(4) COMP.
78 2< 188 10 ih-LOG-THIS-IPC PIC A.
79 2< 189 10 FILLER PIC X.
* Definition from client's perspective: constant in the range [0:2000]

10 uow-air-DATA-BYTE-COUNT PIC 9(4) COMP.

** SIZE OF CLIENT-DATA MUST BE CONSISTENT WITH DATA-BYTE-COUNT **

* For any given application, CLIENT-DATA may be fixed length, e.g.:

02 CLIENT-DATA PIC X(DATA-BYTE-COUNT).

10 uow-air-CLIENT-DATA.

15 uow-air-ELEMENT OCCURS 0 TO 2000 TIMES DEPENDING ON uow-air-DATA-BYTE-COUNT

uow-air-ADD-ITEM-REC-UOW.

10 uow-air-ADD-ITEM-REC-UOW.

01 rsp-air.

* COPY rsp-air-add-item-rec-rsp OF "gcob".

** Definition ADD-ITEM-REC-RSP created on 08/31/84 at 16:45

05 rsp-air-ADD-ITEM-REC-RSP.

10 rsp-air-HDR.

15 rsp-air-SELF-IDENT VALUE "UW".

15 rsp-air-UOW-CODE PIC 9(4) COMP.

88 rsp-air-OK VALUE 136.

88 rsp-air-END-TRANSACTION VALUE 4010.

88 rsp-air-END-ITEM-FOUND VALUE 4035.

88 rsp-air-ITEM-UNALTERABLE VALUE 4041.

88 rsp-air-END-ITEM-ALREADY-EXISTS VALUE 4049.

88 rsp-air-END-ITEM-TOO-LONG VALUE 4085.

88 rsp-air-END-ITEM-INVALID-REC-TYPE VALUE 4046.

88 rsp-air-END-ITEM-INVALID-REC-SEQ-NUM VALUE 4058.

10 rsp-air-RETN-CODE-DETAIL PIC S9(4) COMP.

10 rsp-air-RETN-RECORD-SEQ-NUM PIC S9(4) COMP.

********

* COPY uow-atc-attach-compnt-a01-uow OF "gcob".

05 uow-atc-ATTACH-COMPNT-A01-UOW created on 08/31/84 at 16:47

05 uow-atc-ATTACH-COMPNT-A01-UOW.

10 uow-atc-HDR.

15 uow-atc-SELF-IDENT VALUE "UW".

15 uow-atc-UOW-CODE PIC 9(4) COMP.

10 uow-atc-PARENT-ITEM-ID PIC X(12).

10 uow-atc-COMPNT-ID PIC X(12).

10 uow-atc-REL-POSITION PIC S9(4) COMP.

10 uow-atc-COMPNT-TYPE PIC 9(4) COMP.

* COPY rsp-atc-attach-compnt-a01-rsp OF "gcob".

** Definition ATTACH-COMPNT-A01-RSP created on 08/31/84 at 16:47

05 rsp-atc-ATTACH-COMPNT-A01-RSP.

10 rsp-atc-HDR.

15 rsp-atc-SELF-IDENT VALUE "UW".

15 rsp-atc-UOW-CODE PIC 9(4) COMP.

10 rsp-atc-RETN-RECORD-SEQ-NUM PIC S9(4) COMP.
Figure 8-1. SCREEN COBOL Code for Agent (continued)

PAGE 5 $MERC.B9110.SMPAGNT TANDEM / SCREEN COBOL

2< 716 88 rsp-atc-OK VALUE 0.
2< 717 88 rsp-atc-E-BAD-TRANSACTION VALUE 4010.
2< 718 88 rsp-atc-E-PARENT-NOT-FOUND VALUE 4037.
2< 719 88 rsp-atc-E-COMPNT-NOT-FOUND VALUE 4039.
2< 720 88 rsp-atc-E-COMPNT-CYCLE VALUE 4038.
2< 721 88 rsp-atc-E-ITEM-UNALTERABLE VALUE 4041.
2< 722 88 rsp-atc-E-INVALID-REL-POSITION VALUE 4057.
2< 723 88 rsp-atc-E-INVALID-COMPNT-TYPE VALUE 4096.
2< 724 88 rsp-atc-E-ITEM-TOD-COMPLEX VALUE 4036.
2< 725 10 rsp-atc-RECIP-RETN-CODE-DETAIL PIC 59(4) COMP.
2< 726 10 rsp-atc-INSERTED-POSITION PIC 59(4) COMP.

***********
01 uow-ar-block.

* COPY uow-ar-add-recip-uow OF "gcob".
2< 386 * Definition ADD-RECIP-UOW created on 08/31/84 at 16:45
2< 387 05 uow-ar-ADD-RECIP-UOW.
2< 388 10 uow-ar-HDR.
2< 389 15 uow-ar-SELF-IDENT PIC AA VALUE "UW".
2< 390 15 uow-ar-UOW-CODE PIC 9(4) COMP value 114.
2< 391 10 uow-ar-ITEM-ID.
2< 392 15 uow-ar-DUMMY PIC X(12).
2< 393 10 uow-ar-RECIP-NAME PIC X(120).
2< 394 10 uow-ar-RECIP-TYPE PIC 9(4) COMP.
2< 395 10 uow-ar-OPTIONS.
2< 396 15 uow-ar-USE-DEPOT-RESOL-FLAGS PIC A.
2< 397 15 uow-ar-DEFER-LOCAL-RESOLUTION PIC A.
2< 398 15 uow-ar-DEFER-REMOTE-RESOLUTION PIC A.
2< 399 15 uow-ar-DERIVED-FROM-DLIST PIC A.
2< 400 15 uow-ar-RESERVED-4 PIC A.
2< 401 15 uow-ar-RESERVED-5 PIC A VALUE "N".
2< 402 15 uow-ar-RESERVED-6 PIC A VALUE "N".
2< 403 15 uow-ar-RESERVED-7 PIC A VALUE "N".

* COPY rsp-ar-add-recip-rsp OF "gcob".
2< 410 * Definition ADD-RECIP-RSP created on 08/31/84 at 16:46
2< 411 05 rsp-ar-ADD-RECIP-RSP.
2< 412 10 rsp-ar-HDR.
2< 413 15 rsp-ar-SELF-IDENT PIC AA VALUE "UW".
2< 414 15 rsp-ar-UOW-CODE PIC 9(4) COMP.
2< 415 10 rsp-ar-RETN-CODE PIC 59(4) COMP.
2< 416 88 rsp-ar-OK VALUE 0.
2< 417 88 rsp-ar-W-REC-ALREADY-EXISTS VALUE -4049.
2< 418 88 rsp-ar-E-BAD-TRANSACTION VALUE 4010.
2< 419 88 rsp-ar-E-MUST-BE-YN VALUE 4051.
2< 420 88 rsp-ar-E-RESERVED-MUST-BE-N VALUE 4052.
2< 421 88 rsp-ar-E-RESERVED-MUST-BE-Y VALUE 4053.
2< 422 88 rsp-ar-E-ITEM-NOT-FOUND PIC 59(4) COMP.
2< 423 88 rsp-ar-E-ITEM-NOT-PKG-HDR VALUE 4041.
2< 424 88 rsp-ar-E-ITEM-UNALTERABLE VALUE 4066.
2< 425 88 rsp-ar-W-INVALID-RECIP-TYPE VALUE 4066.
2< 426 88 rsp-ar-W-ITEM-UNKNOWN VALUE -4069.
**Figure 8-1. SCREEN COBOL Code for Agent (Continued)**

```cobol
 Figure 8-1. SCREEN COBOL Code for Agent (Continued)

2< 427  88 rsp-ar-W-REMOTE-NAME-ACCEPTED VALUE '4067.
2< 428  88 rsp-ar-E-RECIPIENT-NAME VALUE '5752.
2< 429  88 rsp-ar-E-RECIPIENT-BAD-NAME VALUE '5754.
2< 430  88 rsp-ar-E-RECIPIENT-BAD-SUFFIX VALUE '5757.
2< 431  88 rsp-ar-E-RECIPIENT-NSRV-FAIL VALUE '5750.
2< 432  88 rsp-ar-E-RECIPIENT-NO-SUCH-NAME VALUE '5751.
2< 433  88 rsp-ar-E-RECIPIENT-SECURITY VALUE '5752.
2< 434  88 rsp-ar-E-RECIPIENT-NSRV-FAIL VALUE '5753.
2< 435  88 rsp-ar-E-RECIPIENT-NSRV-FAIL VALUE '5754.
2< 436  88 rsp-ar-E-RECIPIENT-NSRV-FAIL VALUE '5755.
2< 437  88 rsp-ar-E-RECIPIENT-NSRV-FAIL VALUE '5756.
2< 438  88 rsp-ar-E-RECIPIENT-AMBIGUOUS-NAME VALUE '5757.
2< 439  10 rsp-ar-RETURNCODEDETAIL PIC S9(4) COMP.
2< 440  10 rsp-ar-ACCEPTED-NAME PIC X(120).

************
74
75 01 uow-cri-block.
76 * COPY uow-cri-create-item-uow OF "gcob".
77 * Definition CREATE-ITEM-UOW created on 08/31/84 at 16:48
78 2< 786 05 uow-cri-CREATE-ITEM-UOW.
79 2< 788 10 uow-cri-HDR.
80 2< 789 15 uow-cri-SELF-IDENT PIC AA VALUE "UW".
81 2< 790 15 uow-cri-UOW-CODE PIC 9(4) COMP value 103.
82 2< 791 10 uow-cri-ITEM-TYPE PIC 9(4) COMP.
83 2< 792 10 uow-cri-ITEM-ID.
84 2< 793 15 uow-cri-DUMMY PIC X(12).

************
85
86 01 uow-sp-block.
87 * COPY uow-sp-submit-pkg-uow OF "gcob".
88 * Definition SUBMIT-PKG-UOW created on 08/31/84 at 16:55
89 2< 1966 05 uow-sp-SUBMIT-PKG-UOW.
90 2< 1968 10 uow-sp-HDR.
91 2< 1969 15 uow-sp-SELF-IDENT PIC AA VALUE "UW".
93 2< 1971 10 uow-sp-ITEM-ID.
94 2< 1972 15 uow-sp-DUMMY PIC X(12).
95```

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>Extension</th>
<th>PIC</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2&lt; 1927</td>
<td>88 rsp-usv-E-FLD-NOT-SAME-NODE</td>
<td>VALUE 5697.</td>
<td>59(4)</td>
<td>COMP.</td>
</tr>
<tr>
<td>2&lt; 1928</td>
<td>88 rsp-usv-E-FLD-AMBIGUOUS-NAME</td>
<td>VALUE 5698.</td>
<td>59(4)</td>
<td>COMP.</td>
</tr>
<tr>
<td>2&lt; 1929</td>
<td>10 rsp-usv-RETN-CODE-DETAIL</td>
<td></td>
<td>9(4)</td>
<td></td>
</tr>
<tr>
<td>92</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
93 / LINKAGE SECTION.
94 01 ls-lnk-agent-link.
95 01 ls-lnk-agent-link-reply.
96 99 * COPY ag-lnk-agent-link OF "glnk".
97 99 * COPY ag-rep-agent-link-reply OF "glnk".
98 01 ls-lnk-agent-link-reply.
99 01 ls-lnk-agent-link-reply.
3x 1974 * Definition AGENT-LINK created on 08/31/84 at 17:20
3x 1977 10 AG-LNK-SESSION-ID.
3x 1978 15 AG-LNK-DUMMY PIC X(18).
3x 1978 10 AG-LNK-SENDER-INFO.
3x 1978 15 AG-LNK-SENDER-NAME PIC X(120).
3x 1979 15 AG-LNK-SENDER-APPLIC-ID PIC 9(4) COMP.
3x 1980 10 AG-LNK-RECIPIENT-INFO.
3x 1980 15 AG-LNK-RECIPIENT-NAME PIC X(120).
3x 1981 10 AG-LNK-PACKAGE-INFO.
3x 1981 15 AG-LNK-PACKAGE-ID.
3x 1982 20 AG-LNK-DUMMY PIC X(12).
3x 1983 15 AG-LNK-AGENT-SELECTOR PIC 9(4) COMP.
3x 1984 15 AG-LNK-PACKAGE-FLAGS.
3x 1985 20 AG-LNK-CERTIFIED PIC A.
3x 1986 20 AG-LNK-BYTE REDEFINES AG-LNK-CERTIFIED PIC X.
3x 1987 20 AG-LNK-RESERVED-1 PIC A.
3x 1988 20 AG-LNK-RESERVED-2 PIC A.
3x 1989 20 AG-LNK-RESERVED-3 PIC A.
3x 1990 20 AG-LNK-RESERVED-4 PIC A.
3x 1991 20 AG-LNK-RESERVED-5 PIC A.
3x 1992 20 AG-LNK-RESERVED-6 PIC A.
3x 1993 20 AG-LNK-RESERVED-7 PIC A.
3x 1994 15 AG-LNK-SUBJECT-STRING PIC X(140).
3x 1995 10 AG-LNK-DEPOT-INFO.
3x 1996 15 AG-LNK-AGENT-DATA PIC X(80).
3x 1997 01 ls-rep-agent-link Reply.
3x 1998 01 ls-rep-agent-link Reply.
3x 2000 01 ls-rep-agent-link Reply.
3x 2001 01 ls-rep-agent-link Reply.
3x 2002 01 ls-rep-agent-link Reply.
3x 2003 01 ls-rep-agent-link Reply.
3x 2004 01 ls-rep-agent-link Reply.
3x 2005 15 AG-LNK-AGENT-SELECTOR PIC 9(4) COMP.
3x 2006 15 AG-LNK-PACKAGE-FLAGS.
3x 2007 15 AG-LNK-PACKAGE-FLAGS.
3x 2008 88 AG-LNK-DEFAULT-PACKAGE Value is 0.
3x 2009 20 AG-LNK-CERTIFIED PIC A.
3x 2010 20 AG-LNK-RESERVED-1 PIC A.
3x 2011 20 AG-LNK-RESERVED-2 PIC A.
3x 2012 20 AG-LNK-RESERVED-3 PIC A.
3x 2013 20 AG-LNK-RESERVED-4 PIC A.
3x 2014 20 AG-LNK-RESERVED-5 PIC A.
3x 2015 20 AG-LNK-RESERVED-6 PIC A.
3x 2016 20 AG-LNK-RESERVED-7 PIC A.
3x 2017 15 AG-LNK-AGENT-DATA PIC X(80).
PROCEDURE DIVISION USING ls-link-agent-link, ls-rep-agent-link-reply.

* MAIN ROUTINE

A-MAIN.

* NOTE - This agent assumes that:
  (1) The name of the TISERV Server Class is "TISERV"
  (2) The name of the INBOX folder is "INBOX"
  Use the UDQ "GET-CONFIG-NAME" to avoid making these assumptions.

* (1) SET UP THE IPC HEADER

MOVE "A01" TO ih-version-code.
MOVE ag-lnk-session-id TO ih-session-id.
MOVE "N" TO ih-log-this-ipc.
MOVE 0 TO ih-uows-returned.

* (2) PERFORM SERVICES

IF ag-lnk-agent-data NOT = SPACES PERFORM 100-forward.
PERFORM 200-ack-and-unsave.

* (3) EXIT

PERFORM 800-set-return-code.
PERFORM 900-exit.

100-forward.

* FIRST, CREATE A NEW ITEM (new item-id will be in rsp-cri-item-id)

MOVE 111 TO uow-cri-item-type.
See Appendix B for Item Types.
MOVE "Y" TO uow-cri-is-pkg-hdr.
MOVE 1 TO ih-uows-to-process.
MOVE -3 TO ih-request-code.
The following 2 UOW's will be sent in the same SEND (Inter-
Process Communication). The reason this can be done is
threefold:

1) They all go to the same server class (TISERV).
2) Neither of the UOW's needs information from the other (This
is why the above UOW-CRI could not be included).
3) Each UOW sent has a separate working-storage area (If you
sent 2 UOW-CRI's in the same SEND, they would need
separate working-storage areas).

The advantage of doing this is: fewer messages sent.
The disadvantage of doing this is: longer messages.

BUILD THE UOW TO ADD THE FORWARDEE AS A RECIPIENT

BUILD THE UOW TO ADD A SUBJECT RECORD TO THE NEW ITEM

ISSUE THE SEND
Figure 8-1. SCREEN COBOL Code for Agent (Continued)

```
SEND ih-ipc-hdr, uow-ar-add-recip-uow, uow-air-add-item-rec-uow
TO "TISERV"
REPLY CODE 0, 1, 2 YIELDS
ih-ipc-hdr, rsp-ar-add-recip-rsp, rsp-air-add-item-rec-rsp,
CODE 3 YIELDS ih-ipc-hdr.

**********************************************************************
* The following 3 UOW's will be sent in the same SEND
**********************************************************************

**********************************************************************
* BUILD THE UOW TO REFERENCE THE ORIGINAL ITEM IN THE NEW ITEM
**********************************************************************

MOVE rsp-cri-item-id TO uow-air-item-id.
MOVE ag-lnk-package-id TO uow-air-client-data.
MOVE 102 TO uow-air-rec-type.
* See Appendix B for Rec Types.
MOVE -1 TO uow-air-rec-seq-num.
MOVE 12 TO uow-air-data-byte-count.

**********************************************************************
* BUILD THE UOW TO ADD THE ORIGINAL PACKAGE TO THE NEW PACKAGE
**********************************************************************

MOVE rsp-cri-item-id TO uow-atc-parent-item-id.
MOVE ag-lnk-package-id TO uow-atc-compnt-id.
MOVE 1 TO uow-atc-rel-position.
MOVE 100 TO uow-atc-compnt-type.

**********************************************************************
* BUILD THE UOW TO SUBMIT THE NEW PACKAGE FOR DELIVERY
**********************************************************************

MOVE rsp-cri-item-id TO uow-sp-item-id.

**********************************************************************
* ISSUE THE SEND
**********************************************************************

MOVE 3 TO ih-uows-to-process.
MOVE -2 TO ih-request-code.
SEND ih-ipc-hdr, uow-air-add-item-rec-uow, uow-atc-attach-compnt-a01-uow,
uow-sp-submit-pkg-uow
TO "TISERV"
REPLY CODE 0, 1, 2 YIELDS
Figure 8-1. SCREEN COBOL Code for Agent (Continued)

```cobol
271  ih-ipc-hdr, rsp-air-add-item-rec-rsp, rsp-atc-attach-compnt-a01-rsp,
    rsp-sp-submit-pkg-rsp.
272  CODE 3 YIELDS ih-ipc-hdr.
273
274
275
276  200-ack-and-unsave.
277
278  **********************************************************************
279  * The following 2 UOW's will be sent in the same SEND              *
280  **********************************************************************
281
282  **********************************************************************
283  * BUILD THE UOW TO ACKNOWLEDGE RECEIPT OF THE ORIGINAL PACKAGE     *
284  *
285  *    uow-ack-ack-receipt-uow                                     *
286  *
287  **********************************************************************
288
289  MOVE ag-lnk-package-id TO uow-ack-item-id.
290
291  **********************************************************************
292  * BUILD THE UOW TO REMOVE THE ORIGINAL PACKAGE FROM THE INBOX      *
293  *
294  *    uow-usv-unsave-item-uow                                    *
295  *
296  **********************************************************************
297
298  MOVE ag-lnk-package-id TO uow-usv-item-id.
299  MOVE "INBOX" TO uow-usv-folder-name.
300
301  **********************************************************************
302  * ISSUE THE SEND                                                  *
303  **********************************************************************
304
305  MOVE 2 TO ih-uows-to-process.
306  MOVE -2 TO ih-request-code.
307
308  SEND ih-ipc-hdr, uow-ack-ack-receipt-uow uow-usv-unsave-item-uow,
    TO "TISERV".
309  REPLY CODE 0, 1, 2 YIELDS
    ih-ipc-hdr.
310
311  SEND ih-ipc-hdr, rsp-ack-ack-receipt-rsp, rsp-usv-unsave-item-rsp,
    TO "TISERV".
312  REPLY CODE 3 YIELDS ih-ipc-hdr.
313
314
315
316  800-set-return-code.
317
318  MOVE 0 TO ag-rep-error-return.
319  MOVE SPACES TO ag-rep-error-msg.
320
321
322  900-exit.
323
324  EXIT PROGRAM.
```
Figure 8-1. SCREEN COBOL Code for Agent (Continued)
The clients and agents that operate in a TRANSFER application have distinguishable characteristics:

- Clients are processes that operate within their own area of control. They manage their own transactions.

- Agents function as logical subroutines operating inside TRANSFER transactions. They temporarily take over the TRANSFER area of control. Agents—even server agents operating as separate processes—always complete their operations before returning control to TRANSFER.

Queue management allows an agent to schedule an interaction between a client and the TRANSFER data base. Scheduling is an important consideration because many clients need to acquire resources outside of TRANSFER in order to carry out their interactions with the TRANSFER data base. With scheduling, the agent does not have to wait for the client to obtain the resources and perform the interaction and, consequently, causes minimum interruption of TRANSFER activities. The client waits for appropriate work.

An agent can enqueue an event to the queue manager after possibly performing some delivery related operations such as saving the package in a special folder. Whenever the client has the required resources outside of the TRANSFER environment, it can request work from the queue manager. This interaction is illustrated in Figure 9-1.

Queue management is not restricted to use with clients and agents. The queueing function can be used in any application that needs to separate an activity into multiple stages.
Activities of Client A are initiated by an event outside of TRANSFER; this interaction presents no problem.

Activities of Client B are initiated by an event within TRANSFER; without the queue manager, Client B would have to be ready for such an initiation and TRANSFER would have to wait for the agent/client interaction to complete.

Figure 9-1. Interaction Between TRANSFER Clients

**QUEUE MANAGER**

The queue manager provides a facility where queue entries can be supplied and consumed by different requesters. Throughout this section the term supplier refers to the process that places an entry on the queue; the term consumer refers to the process that retrieves an entry from the queue for processing.

The queue manager is passive in that the consumption of queue entries is initiated strictly by consumer request and performed by external instructions.

The queue manager consists of two server types:

- **Entry Manager** handles enqueuing, reading, and dequeuing of queue entries.
- **Wait Manager** handles waiting for new queue entries.

The Entry Manager and Wait Manager servers operate on a common file called the Queue file; this file can contain information for many different queues. When you configure the server classes in a PATHWAY environment, you can have many Entry Managers per Queue file, but only one Wait Manager per Queue file.
The Queue file is an audited key-sequenced file that has been previously created and in which multiple queues can be stored. Each queue entry has a 32-byte queue-name field, a priority field, a timestamp field, a cpu-pin field that holds the cpu,pin of the server placing the entry on the queue, and a user data field as appropriate.

The maximum length for records in the Queue file controls the amount of user data that can be present in each queue entry. The queue manager reserves 128 bytes at the beginning of the physical record; user data can occupy the remaining space in the record.

The system manager can partition the Queue file by the first two bytes of the physical key. To facilitate this partitioning, the first two bytes of the queue-name field are used as the first two bytes of the Queue file physical key.

The queue manager provides four functions for a queue:

- Dequeue an entry from the queue (DEQ) Requests are sent to an Entry Manager
- Enqueue an entry on the queue (ENQ)
- Read an entry from the queue (READQ) Requests are sent to the Wait Manager
- Wait for an entry on the queue (WAITQ)

These functions are supported via a message interface suitable for use from SCREEN COBOL.

Dequeu (DEQ) Function

A dequeue function, which is initiated by a consumer, is handled by a DEQ request that must take place in transaction mode. The DEQ request results in a READLOCK followed by a WRITEUPDATE to delete the entry from the queue.

The function does not require an END-TRANSACTION to be issued soon after the DEQ request. The transaction can persist because a lock on a deleted record does not block read requests to other records with a similar key value.
Queue Management

Enqueue (ENQ) Function

An enqueue function, which is initiated by a supplier, is handled by an ENQ request that must take place in transaction mode. An entry is inserted on the queue in a single WRITE. The new insertion remains locked until an END-TRANSACTION is issued; the lock will block any read request on that record and this could prevent DEQ operations from succeeding on available queue entries with a similar key value. Lockout can be prevented by including only a minimum of application work with the ENQ request within the TMF transaction.

The ENQ request causes a message to be sent to the Wait Manager for the Queue file. This message allows the Wait Manager to respond to a WAITQ request.

Read (READQ) Function

A read function, which might be initiated by either a supplier or a consumer, is handled by a READQ request that does not need to take place in transaction mode. The function retrieves an entry from the queue but does not delete the entry. This allows applications to place information on a queue and access it repeatedly in read-only mode.

Wait (WAITQ) Function

A wait function, which is initiated by a consumer, is handled by a WAITQ request that should not take place in transaction mode. The wait could be extensive, and transactions should not be outstanding for long periods of time.

When a WAITQ request is received for a particular queue, the Wait Manager performs a read to ensure there is no entry on that queue. If an entry is on the queue, the Wait Manager issues a response to the consumer; if the queue is empty, the Wait Manager waits for an entry. The application logic should attempt a DEQ before calling WAITQ if it typically expects other entries. If other entries are not expected, the WAITQ should only be issued immediately after completing a DEQ.

USING THE QUEUE MANAGER

Figure 9-2 illustrates the DEQ, ENQ, and READQ UOWs being sent to the Entry Manager and the WAITQ UOW being sent to the Wait Manager.
The large arrows show the effective direction of control. An agent or client requests services of the Entry Manager while the Wait Manager triggers a waiting client that has previously issued a WAITQ request.

**Figure 9-2. Queue Management Interaction**
Queue Management

In the TRANSFER environment, the UOWs would typically be used in the following sequence:

**Step 1**

Upon receipt of a package, an agent would issue an ENQ request to the Entry Manager. The ENQ would be a request for services of the client handling that particular type of package. The agent would probably move the package to a folder other than INBOX where the client would expect to find it. The agent returns, its role completed.

BEGIN-_TRANSACTION issued by TAREQ for delivery to correspondents at this node

Invoke AGENT during delivery to one correspondent

SEND SAVE-ITEM and UNSAVE-ITEM UOWs to TISERV.

SEND ENQ UOW to Entry Manager.

END-_TRANSACTION issued by TAREQ.

**Step 2**

The client would first issue a WAITQ request to the Wait Manager, specifying it wants to wait until a specific type of request is placed in the queue. This request will be issued outside of transaction mode because the client could be waiting a long time to receive a reply from the Wait Manager.

**Step 3**

When an entry is placed in the queue for the type requested, the client will receive a reply indicating the queue entry to process. The client will then retrieve the entry by issuing a DEQ UOW to the Entry Manager in transaction mode. The TMF transaction is now under the control of the client and not automatically handled on behalf of the application as it was for an agent.
SEND a WAITQ UOW to the Wait Manager.
BEGIN-TRANSACTION.
SEND a DEQ UOW to the Entry Manager.
    Process the queue entry.
END-TRANSACTION.

The preceding example assumes that typically there would be no entry in the queue for the client to process. If the application expected an entry to be waiting in the queue for the client, it should first issue the DEQ request before issuing the WAITQ request, as shown in the following example.

BEGIN-TRANSACTION.
SEND a DEQ UOW to the Entry Manager.
If the queue contains an entry of the desired type,
    process the queue entry and
    END-TRANSACTION and
----- look for more entries
Else
    END-TRANSACTION and
    SEND a WAITQ UOW to the Wait Manager.

The program would normally be in the DEQ and process loop, entering the WAITQ loop only when the DEQ ran out of entries to process.

READQ is typically used to read entries in a queue for a status report.
Queue Management

SERVER INTERFACE

Processes interface with the Wait Manager and the Entry Manager servers through units-of-work (UOWs) issued within requests. The format of requests issued by a program is the same, whether the request is issued by a SEND statement in SCREEN COBOL or by a WRITEREAD call in TAL. The data buffer for each request consists of the following:

- A request header that specifies version information and error handling controls and that provides space for reply and return codes.
- One or more UOWs, each of which specifies a code for an operation plus any necessary parameters. For example, the DEQ UOW request would include the name of the queue from which an entry is to be removed.

The reply from a server, which is written in the same format as the request to which it responds, consists of the following:

- A reply header, with reply and return codes supplied by the server.
- One or more response UOWs that return requested data or completion status to the consumer. For example, the response to a DEQ UOW request would include the data of the queue entry removed from the queue.

For each UOW in the request there is a corresponding response UOW in the overall reply. The servers process UOWs in the order in which they appear in the request. Thus, the response UOWs are returned in the same order as the corresponding UOWs in the request.

For certain errors, however, no response UOWs are included. The number of UOWs in the reply is indicated by a field in the reply header. A response UOW will never be skipped in the reply; the first error that would cause a UOW to be excluded from the reply will terminate the request.

Each UOW has a standard header that identifies the operation requested. Each response UOW has a return code that identifies the action taken in response to the corresponding request UOW.
Queue Management

Request and Reply Headers

The format for request and reply headers is the same for all requests and replies. Within an application program, these headers are defined together as the interprocess communication (IPC) header. Your application must establish values for the IPC header fields that are transmitted in the request.

All requests to and replies from the Entry Manager and Wait Manager servers must be prefixed with an IPC header. The basic format for the messages conforms to the TRANSFER IPC format. The following rules apply:

- You must supply an IPC header followed by one or more UOWs.
- Each UOW must be a DEQ, ENQ, READQ, or WAITQ UOW.
- The WAITQ UOW must be the only UOW in an IPC request. If you specify more than one UOW for the Wait Manager, the PW-REPLY-CODE field of the IPC-HDR will be set to RQST-ERR and the IPC-RETN-CODE field will be set to RQST-TOO-LONG.

IPC Header Format

The format of the IPC header is shown by the following DDL definition:

```plaintext
DEF ipc-hdr.
  02 request-code              TYPE BINARY 16.
    88 stop-on-warning         VALUE -1.
    88 stop-on-err             VALUE -2.
    88 do-all-uows             VALUE -3.
  02 pw-reply-code             TYPE BINARY 16
    REDEFINES REQUEST-CODE.
    88 all-uows-ok             VALUE 0.
    88 uows-with-warning       VALUE 1.
    88 uows-with-err           VALUE 2.
    88 rqst-err                VALUE 3.
  02 filler                   PIC X.
  02 version-code.            PIC A.
    03 letter                  PIC A.
    03 rev-number              PIC 99.
  02 ipc-retn-code            TYPE BINARY 16 VALUE 0.
    88 ipc-ok                 VALUE 0.
    88 invalid-version-code    VALUE 1.
    88 invalid-session-id     VALUE 2.
```

Individual fields in the IPC header can contain the following information:

- **REQUEST-CODE**

  In a request to the Entry Manager or Wait Manager server, your application sets this field to indicate request processing conditions. In the request, this field always contains a value less than zero, as follows.

  **STOP-ON-WARNING** (-1) stops the processing if a UOW warning indication is encountered. Warnings imply successful completion of the UOW in which they occur.

  **STOP-ON-ERR** (-2) stops the processing if a UOW error occurs. Errors imply that the UOW was not processed successfully.

  **DO-ALL-UOWS** (-3) requests processing of all UOWs in the request. Processing halts only if a request error, as defined by the RQST-ERR value, or a system error occurs.

- **PW-REPLY-CODE**

  In a reply from the server, the REQUEST-CODE field is redefined as the PW-REPLY-CODE field. This field contains a value that indicates request processing results. In the reply, this field always contains a value of zero or greater, as follows:

  **ALL-UOWS-OK** (0) indicates that all UOWs in the request were processed successfully.
UOWS-WITH-WARNING (1) indicates that warning indications were encountered in one or more UOWs. If STOP-ON-WARNING was not specified in the REQUEST-CODE field, all UOWs in the request have a corresponding response UOW in the reply. If STOP-ON-WARNING was specified, only those UOWs preceding and including the first with a warning indication have corresponding response UOWs.

UOWS-WITH-ERR (2) indicates that errors were detected in one or more UOWs. If neither STOP-ON-ERR nor STOP-ON-WARNING was specified in the REQUEST-CODE field, all UOWs in the request have a corresponding response UOW. If STOP-ON-ERR or STOP-ON-WARNING was specified, only those UOWs preceding and including the first with an error have corresponding response UOWs. If STOP-ON-ERR is specified in the request, the UOWs preceding the first with an error might return warning indications.

RQST-ERR (3) indicates that a request error occurred. This type of error typically indicates that something was wrong with the data in the IPC header and that the error is not specific to any particular UOW. In certain cases, the error might involve an individual UOW—for example, one with an invalid UOW header. For further information, your application should examine the IPC-RETN-CODE field.

CAUTION

If your application receives a value of 2 or 3 in this field, the transaction should be aborted, causing transaction backout. If the transaction is not backed out, consistency in the TRANSFER data base cannot be guaranteed.

- VERSION-CODE

This field designates the version of the message formats used. The version code is defined by Tandem and consists of a letter followed by a two-digit revision number.

- IPC-RETN-CODE

In a request to the server, this field is ignored. In a reply from the server, one of the following values appears.

IPC-OK (0) indicates the server detected no errors in the IPC header. Warning indications or errors, however, might have been present in the individual UOWs in the request.
INVALID-VERSION-CODE (1) indicates the request contained an
IPC version code that could not be recognized by the server.

INVALID-SESSION-ID (2) refers to access validation in
TRANSFER servers and is not applicable to queue management.

SERVICE-DENIED (3) is reserved for use by Tandem.

INVALID-UOW-HDR (4) indicates an invalid UOW header appeared
in your request. This could occur if the previous UOW
specified the wrong size or if you omitted the value UW from
the UOW header.

RQST-TOO-LONG (5) indicates the request was too long; that
is, the request contained more UOWs than the number
specified in the UOWS-TO-PROCESS field, contained more data
than the buffer allowed, or contained more than one WAITQ
UOW. Buffer size is determined by the MAXREPLY and
MAXREQUEST parameters when configuring PATHWAY or when
starting the servers.

REPLY-TOO-LONG (6) indicates the reply was too long; the
buffer space allotted for the reply was insufficient.

RQST-TOO-SHORT (7) indicates the request was too short; that
is, the request contained fewer UOWs than the number
specified in the UOWS-TO-PROCESS field, or fewer bytes were
sent than were expected.

INVALID-REQUEST-CODE (8) indicates the REQUEST-CODE field
contained an invalid entry.

E-BAD-TRANSACTION (4010) indicates that the request did not
have a TMF transaction and attempted to perform an operation
that required one, or that the transaction associated with
the request is unusable, probably due to the failure of some
component of the system or network. IPC-RETN-CODE-DETAIL
contains the actual GUARDIAN file error code.

E-ERR-QUEUE-FILE (4924) indicates an error occurred on the
Queue file. Check the IPC-RETN-CODE-DETAIL.

E-IO-TIMEOUT (4990) indicates a timeout occurred on file
I/O. This usually indicates a deadlock with another
process. IPC-RETN-CODE-DETAIL contains the file code of the
file on which the error occurred. If you receive this error
and the request was issued under a TMF transaction, you must
abort the TMF transaction; you should then retry the request
under a new transaction.
Queue Management

E-WAITMANAGER-UNAVAIL (6006) indicates an error occurred when the Entry Manager was communicating with the Wait Manager. Applications can use the NOTIFY-WAIT-MANAGER flag in the ENQ UOW to add the queue entry if the application is willing to make the entry when the Wait Manager is unavailable.

- **IPC-RETN-CODE-DETAIL**

  In a request to the server, this field is ignored. In a reply from the server, the meaning of this field depends on the value of IPC-RETN-CODE as specified for relevant IPC-RETN-CODEs.

- **SESSION-ID**

  This field is used for access validation in TRANSFER servers; the field is ignored for queue management. The field can have any value. The value sent in the request will be returned in the response, thus allowing the same buffer to be used for the IPC header when using queue management and TISERV.

- **UOWS-TO-PROCESS**

  In a request to the server, this field indicates the number of UOWs transmitted with the request. The value is echoed in a reply.

- **UOWS-RETURNED**

  In a request to the server, this field is ignored. In a reply, the field is set to the number of UOWs processed by the server.

- **LOG-TTHIS-IPC**

  In a request to the server, this field specifies whether the request and its reply are logged (Y) or not logged (N).

**UOW HEADER AND RETURN CODE**

Information is transmitted to the Entry Manager and Wait Manager servers through UOWs issued within requests. Information transmitted by a UOW is moved into the individual fields of the UOW by your program. Information returned in a UOW response is entered in the individual fields of the response by the servers.
Queue Management

Each UOW transmitted to the servers begins with a UOW header. The DDL definition for this header is:

```
DEF uow-hdr.
  02 self-ident        PIC AA VALUE "UW".
  02 uow-code          TYPE BINARY 16 UNSIGNED.
END.
```

Individual fields in the UOW header contain the following information:

- **SELF-IDENT** always contains the characters UW to identify the header as a UOW header.
- **UOW-CODE** is a code value that identifies the specific UOW request being made.

Each response UOW returned from the servers begins with the same header as the request, followed by two fields dealing with return codes.

Individual fields in the response UOW header contain the following additional information:

- **UOW-RETN-CODE** is the return code.
  - If no errors were encountered, this field is set to 0.
  - If an error occurred, the field contains a positive value; a positive value indicates the UOW was not processed.
  - If a warning was indicated, the field contains a negative value; a negative value indicates the UOW was successfully processed.

In the RETN-CODE field for each UOW, entries that begin with E denote errors returned to your process by the server, and entries that begin with W denote warnings. All possible entries are listed for each UOW. These errors are summarized at the end of this section.

- **RETN-CODE-DETAIL** is a code that primarily identifies an error detected by a subsystem other than the queue management facility, such as the GUARDIAN operating system or the EXPAND network software, and for which the queue management facility provides no standard handling. These errors are discussed further in Appendix A.

All request and response UOWs must be aligned on word boundaries.
SOFTWARE PROVIDED WITH QUEUE MANAGEMENT

The TRANSFER software release provides three files for interfacing with the queue manager: GPQCOB, GPQLNK, and GPQDDL. These files contain source code for commonly used queue manager elements whose field and structure definitions appear in this manual. Typical queue manager elements are the interprocess communication (IPC) header that initiates a request, and the unit-of-work (UOW) definitions that describe operations to be performed.

This code can be copied into a SCREEN COBOL or COBOL source program by coding the record level and then using the COPY statement

COPY copy-text OF "filename"

where

copy-text is the unique name for the definition in the named file. You determine the correct copy-text name by searching the files.

GPQCOB - This file contains COBOL source code for Working-Storage Section definitions.

GPQLNK - This file contains the same code as the GPQCOB file, but without the INITIAL-VALUE clauses. The GPQLNK file is used for Linkage Section definitions.

GPQCOB and GPQLNK have information in addition to the information produced by DDL when generating COBOL. The primary additions are prefixes for all field names, and many level 88 declarations for fields.

GPQDDL - This file contains DDL code that is used to create definitions for use when programming in a language other than SCREEN COBOL or COBOL.

GPQDDL code does not exactly match the DDL syntax shown in this manual, but it should be immediately obvious how to interpret any differences.

UOW DEFINITIONS

Four UOWs are available for use with the queue management facility. The UOWs and the functions they perform are summarized in Table 9-1.
Queue Management

Table 9-1. Queue Management UOW Summary

<table>
<thead>
<tr>
<th>UOW</th>
<th>UOW Code</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEQ</td>
<td>501</td>
<td>Dequeues an entry from a queue</td>
</tr>
<tr>
<td>ENQ</td>
<td>502</td>
<td>Enqueues an entry on a queue</td>
</tr>
<tr>
<td>READQ</td>
<td>504</td>
<td>Reads an entry from a queue</td>
</tr>
<tr>
<td>WAITQ</td>
<td>505</td>
<td>Waits for an entry on a queue</td>
</tr>
</tbody>
</table>

The following paragraphs present the UOWs. Each definition begins with the DDL format for the UOW request and corresponding response, followed by a description of the fields and the operations performed.

Error codes are listed and defined in Appendix A.
DEQ (UOW Code 501)

DEQ dequeues an entry from a queue. This operation requires a TMF transaction.

**DEF deq-uow.**

```plaintext
02 hdr.
   03 self-ident   PIC AA VALUE "UW".
   03 uow-code    PIC 9(4) COMP VALUE 501.
02 flags.
   03 specific-deq TYPE BOOLEAN VALUE "N".
   03 reserved-1  TYPE BOOLEAN VALUE "N".
   03 reserved-2  TYPE BOOLEAN VALUE "N".
   03 reserved-3  TYPE BOOLEAN VALUE "N".
   03 reserved-4  TYPE BOOLEAN VALUE "N".
   03 reserved-5  TYPE BOOLEAN VALUE "N".
   03 reserved-6  TYPE BOOLEAN VALUE "N".
   03 reserved-7  TYPE BOOLEAN VALUE "N".
   02 queue-name  PIC X(32) VALUE SPACES.
   02 priority   PIC 9(3) COMP VALUE 0.
   02 time-of-enq PIC X(8) VALUE LOW-VALUES.
   02 cpu-pin    PIC X(2) VALUE LOW-VALUES.
   02 max-data-size PIC 9(5) COMP.
   02 pad-char   PIC X(1) VALUE LOW-VALUE.
   02 filler     PIC X(1) VALUE LOW-VALUE.
END.
```

**DEF deq-rsp.**

```plaintext
02 hdr.
   03 self-ident   PIC AA VALUE "UW".
   03 uow-code    PIC 9(4) COMP VALUE 501.
02 retn-code   PIC S9(4) COMP.
02 retn-code-detail PIC S9(4) COMP.
02 queue-name  PIC X(32) VALUE SPACES.
02 priority   PIC 9(3) COMP.
02 time-of-enq PIC X(8) VALUE LOW-VALUES.
02 cpu-pin    PIC X(2) VALUE LOW-VALUES.
02 data-len   PIC 9(5) COMP.

* Applications should define the actual format of the data portion, for example:
* 02 data-field   PIC X(!queue length!).

END.
```
Queue Management

DEQ

DEQ FIELDS. The fields defined in this UOW are:

- HDR is the UOW header. The UOW-CODE value is 501.
- FLAGS lets you qualify the search for a queue entry. Fields within FLAGS are as follows:
  
  SPECIFIC-DEQ specifies whether or not the PRIORITY, TIME-OF-ENQ, and CPU-PIN fields are to be included in the search for the queue entry.

  Y = Include the PRIORITY, TIME-OF-ENQ, and CPU-PIN fields. The search will return the entry requested provided the entry exists.

  N = Do not include the PRIORITY, TIME-OF-ENQ, and CPU-PIN fields. The search will return the entry with the highest priority. If more than one entry has the same PRIORITY, the search will return the entry with the oldest TIME-OF-ENQ. If more than one entry has the same PRIORITY and TIME-OF-ENQ, the search will return the entry with the lowest CPU-PIN.

  This field is typically set to N and not used. The field is expected to be used only when a supplier removes a queue entry that it added; presumably the application no longer needs the entry. When the field is set to N, the PRIORITY field only is checked for validity.

  RESERVED-1 through RESERVED-7 are reserved for future use; these fields must be set to N.

- QUEUE-NAME is the name of the queue from which the entry is to be dequeued. Input to this field is echoed in the response.

- PRIORITY in the request is ignored if SPECIFIC-DEQ = N; the field is checked for validity, however.

  In the request, PRIORITY is the priority value to be used in the search.

  In the response, PRIORITY is the priority value that was specified by the supplier when it made the entry on the queue.

- TIME-OF-ENQ in the request is ignored if SPECIFIC-DEQ = N.

  In the request, TIME-OF-ENQ is timestamp value to be used in the search.

  In the response, TIME-OF-ENQ is the timestamp generated by the Entry Manager that made the entry on the queue.
Queue Management
DEQ

- CPU-PIN in the request is ignored if SPECIFIC-DEQ = N.

In the request, CPU-PIN is the cpu,pin value to be used in the search.

In the response, CPU-PIN is the cpu,pin of the Entry Manager that made the entry.

- MAX-DATA-SIZE is the maximum number of bytes to be returned.

- PAD-CHAR is the character to be used for padding the reply buffer when the actual data length is less than MAX-DATA-SIZE.

- RETN-CODE is the return code. The server returns a code in this field to indicate one of the following entries.

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OK</td>
</tr>
<tr>
<td>4010</td>
<td>E-BAD-TRANSACTION</td>
</tr>
<tr>
<td>4051</td>
<td>E-MUST-BE-YN</td>
</tr>
<tr>
<td>4052</td>
<td>E-RESERVED-MUST-BE-N</td>
</tr>
<tr>
<td>4078</td>
<td>E-INVALID-PRIORITY</td>
</tr>
<tr>
<td>4087</td>
<td>E-INVALID-MAX-DATASIZE</td>
</tr>
<tr>
<td>4089</td>
<td>W-DATA-TRUNCATED</td>
</tr>
<tr>
<td>4924</td>
<td>E-ERR-QUEUE-FILE</td>
</tr>
<tr>
<td>4990</td>
<td>E-IO-TIMEOUT</td>
</tr>
<tr>
<td>6001</td>
<td>W-QUEUE-EMPTY</td>
</tr>
</tbody>
</table>

- RETN-CODE-DETAIL is an error number returned by a subsystem other than the queue management facility or is a further qualification of an error detected by the server.

- DATA-LEN is the actual length of the data field before padding the reply buffer to MAX-DATA-SIZE bytes in length.

- The format of the data in the queue entry returned by DEQ must be defined by the application. This data would immediately follow the DATA-LEN field and be MAX-DATA-SIZE bytes. The maximum size for the data portion of the response is constrained by the record size defined for the Queue file at configuration time. Refer to the TRANSFER Delivery System Management and Administration Guide.

A sample format for the data portion would be:

```
02 data-field PIC X(100).
```

DEQ OPERATION. DEQ retrieves, thereby removing, an entry from the named queue, and returns either an entry from the queue or an error. The error could be a warning that there was nothing on the queue. The application should follow the DEQ operation with a WAITQ request for the case where the queue does not contain an entry for the consumer; the WAITQ request is used as a delay until the queue has an entry.

The response will always contain MAX-DATA-SIZE bytes of data immediately following the DATA-LEN field. If the actual length of the data for the queue entry is less than MAX-DATA-SIZE, or if
Queue Management
DEQ

an error occurs during processing, PAD-CHAR will be used to pad
the field.

The entry returned will be the oldest entry among those with the
highest priority or will be the one specified if SPECIFIC-DEQ is
set to Y.
ENQ (UOW Code 502)

ENQ enqueues an entry on a queue. This operation requires a TMF transaction.

```
DEF enq-uow.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code PIC 9(4) COMP VALUE 502.
  02 flags.
    03 notify-wait-manager TYPE BOOLEAN VALUE "Y".
    03 reserved-1 TYPE BOOLEAN VALUE "N".
    03 reserved-2 TYPE BOOLEAN VALUE "N".
    03 reserved-3 TYPE BOOLEAN VALUE "N".
    03 reserved-4 TYPE BOOLEAN VALUE "N".
    03 reserved-5 TYPE BOOLEAN VALUE "N".
    03 reserved-6 TYPE BOOLEAN VALUE "N".
    03 reserved-7 TYPE BOOLEAN VALUE "N".
  02 queue-name PIC X(32) VALUE SPACES.
  02 priority PIC 9(3) COMP VALUE 0.
  02 data-byte-count PIC 9(5) COMP VALUE 0.

* Applications should define the actual format of the data portion, for example:
  * 02 data-field PIC X(!queue length!).

END.
```

```
DEF enq-rsp.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code PIC 9(4) COMP VALUE 502.
  02 retn-code PIC S9(4) COMP.
  02 retn-code-detail PIC S9(4) COMP.
  02 time-of-enq PIC X(8).
  02 cpu-pin PIC X(2).

END.
```

**ENQ FIELDS**. The fields defined in this UOW are:

- HDR is the UOW header. The UOW-CODE value is 502.
Queue Management

ENQ

- **FLAGS** lets you qualify the entry insertion. Fields within FLAGS are as follows:

  NOTIFY-WAIT-MANAGER specifies whether or not the Wait Manager should be notified about this new queue entry.

  Y = Notify the Wait Manager.

  N = Do not notify the Wait Manager.

  The option not to notify the Wait Manager exists only to allow reduced overhead for applications that do not need the Wait Manager. These applications must have some other mechanism for deciding when to issue a DEQ request.

  This field should always be set to Y unless the Wait Manager is not being used.

  RESERVED-1 through RESERVED-7 are reserved for future use; these fields must be set to N.

- **QUEUE-NAME** is the name of the queue on which the entry is to be inserted. The name is selected by the user.

- **PRIORITY** affects the order in which entries are inserted on the queue for each value of QUEUE-NAME. The field can contain an unsigned value ranging from 0 (lowest priority) through 199 (highest priority).

  Higher numbers will be processed (with a DEQ or READQ request) before lower numbers.

- **DATA-BYTE-COUNT** is the number of data bytes that are present in the request.

  The format of the data in the queue entry supplied by ENQ should be defined by the application. This data would immediately follow the DATA-BYTE-COUNT field and must be DATA-BYTE-COUNT bytes. The maximum size for the data portion of the request is constrained by the record size defined for the Queue file at configuration time. Refer to the TRANSFER Delivery System Management and Administration Guide.

  A sample format for the data portion would be:

  02 data-field PIC X(100).

- **RETN-CODE** is the return code. The server returns a code in this field to indicate one of the following entries.
Queue Management

ENQ

OK
4010  E-BAD-TRANSACTION  4085  E-DATA-TOO-LONG
4051  E-MUST-BE-YN       4924  E-ERR-QUEUE-FILE
4052  E-RESERVED-MUST-BE-N 4990  E-IO-TIMEOUT
4078  E-INVALID-PRIORITY  6006  E-WAITMANAGER-UNAVAIL

- RETN-CODE-DETAIL is an error number returned by a subsystem other than the queue management facility or is a further qualification of an error detected by the server.

- TIME-OF-ENQ is the timestamp generated by the Entry Manager that makes the entry.

- CPU-PIN is the cpu,pin of the Entry Manager that makes the queue entry. This value is used to ensure that entries on the queue are unique.

ENQ OPERATION. ENQ adds an entry to the named queue. The application making the entry should end its transaction as soon as possible after adding the entry. This will minimize conflict when available queue entries exist in the file but are inaccessible because they are locked. If a queue entry remains locked, an available consumer could issue a DEQ, fail, and go back to waiting. The Wait Manager, however, will not be notified that the lock has been removed, forcing the potential consumer to wait until the next entry for QUEUE-NAME is made.

The ordering for queue entries with the same queue-name and priority is based on a combination of the current system timestamp and the cpu-pin of the Entry Manager. These generated values are returned in the response.
Queue Management

READQ

READQ (UOW Code 504)

READQ reads an entry from a queue. This operation does not require a TMF transaction.

```
DEF readq-uow.
  02 hdr.
    03 self-ident          PIC AA VALUE "UW".
    03 uow-code            PIC 9(4) COMP VALUE 504.
  02 flags.
    03 any-queue-name      TYPE BOOLEAN VALUE "N".
    03 any-priority        TYPE BOOLEAN VALUE "Y".
    03 any-time-of-enq     TYPE BOOLEAN VALUE "Y".
    03 any-cpu-pin         TYPE BOOLEAN VALUE "Y".
    03 skip-exact          TYPE BOOLEAN VALUE "N".
    03 reposition          TYPE BOOLEAN VALUE "N".
    03 reserved-6          TYPE BOOLEAN VALUE "N".
    03 reserved-7          TYPE BOOLEAN VALUE "N".
  02 queue-name           PIC X(32) VALUE SPACES.
  02 priority             PIC 9(3) COMP VALUE 0.
  02 time-of-enq          PIC X(8) VALUE LOW-VALUES.
  02 cpu-pin              PIC X(2) VALUE LOW-VALUES.
  02 max-data-size        PIC 9(5) COMP.
  02 pad-char             PIC X(1) VALUE LOW-VALUE.
  02 filler               PIC X(1) VALUE LOW-VALUE.
END.

DEF readq-rsp.
  02 hdr.
    03 self-ident          PIC AA VALUE "UW".
    03 uow-code            PIC 9(4) COMP VALUE 504.
  02 retn-code            PIC S9(4) COMP.
  02 retn-code-detail     PIC S9(4) COMP.
  02 queue-name           PIC X(32) VALUE SPACES.
  02 priority             PIC 9(3) COMP VALUE 0.
  02 time-of-enq          PIC X(8).
  02 cpu-pin              PIC X(2).
  02 data-len             PIC 9(5) COMP.

* Applications should define the actual format of the data
* portion, for example:
* 02 data-field          PIC X(!queue length!).
END.
```
**READQ** FIELDS. The fields defined in this UOW are:

- **HDR** is the UOW header. The UOW-CODE value is 504.
- **FLAGS** lets you qualify the read operation. Fields within **FLAGS** are as follows.

  ANY-QUEUE-NAME specifies whether or not the read is restricted to a specific queue.

  \[ Y = \text{The read is not restricted to a specific queue. A } Y \text{ value in this field will cause ANY-PRIORITY, ANY-TIME-OF-ENQ, and ANY-CPU-PIN to be automatically set to } Y. \]

  \[ N = \text{The read is restricted to the queue name entered in the QUEUE-NAME field of the request.} \]

  ANY-PRIORITY specifies whether or not the read is restricted to a specific value of PRIORITY.

  \[ Y = \text{The read is not restricted to a specific priority. A } Y \text{ value in this field will cause ANY-TIME-OF-ENQ and ANY-CPU-PIN to be automatically set to } Y. \]

  \[ N = \text{The read is restricted to the priority entered in the PRIORITY field of the request.} \]

  ANY-TIME-OF-ENQ specifies whether or not the read is restricted to a specific value of TIME-OF-ENQ.

  \[ Y = \text{The read is not restricted to a specific timestamp. A } Y \text{ value in this field will cause ANY-CPU-PIN to be automatically set to } Y. \]

  \[ N = \text{The read is restricted to the timestamp entered in the TIME-OF-ENQ field of the request.} \]

  ANY-CPU-PIN specifies whether or not the read is restricted to a specific value of CPU-PIN.

  \[ Y = \text{The read is not restricted to a specific cpu and pin.} \]

  \[ N = \text{The read is restricted to the cpu and pin entered in the CPU-PIN field of the request.} \]
Queue Management

READQ

SKIP-EXACT determines which entry is to be read.

Y = Read and return the entry immediately following the entry that was an exact match of the specified fields.

N = Read and return the entry that is an exact match of the specified fields.

REPOSITION is used for sequential reading of a specific queue or the entire Queue file. Setting this flag to Y after the initial read informs the Entry Manager that the values in the four fields QUEUE-NAME, PRIORITY, TIME-OF-ENQ, and CPU-PIN are being supplied based on the result of a previous READQ request.

To read sequentially from the beginning of a specified queue or from the beginning of the Queue file:

- Let SKIP-EXACT and REPOSITION = N
- Set the four ANY- flags = as desired
- Set ANY- flags' corresponding fields = as appropriate

For subsequent sequential reads:

- Set SKIP-EXACT and REPOSITION = Y
- Let ANY- flags remain at their same settings.
- Copy all fields from the previous READQ response into the UOW for the next request.

An end-of-file warning indicates the end of the group has been reached.

RESERVED-6 and RESERVED-7 are reserved for future use; these fields must be set to N.

- QUEUE-NAME in the request is the name of the queue to which the read is restricted.

QUEUE-NAME in the response is the name of the queue from which the entry was taken.

- PRIORITY in the request is the priority value to which the read is restricted.

PRIORITY in the response is the value specified by the application that made the queue entry.
Queue Management
READQ

- **TIME-OF-ENQ** in the request is the Entry Manager timestamp to which the read is restricted.

  TIME-OF-ENQ in the response is the timestamp generated by the Entry Manager that made the queue entry.

- **CPU-PIN** in the request is the Entry Manager cpu,pin to which the read is restricted.

  CPU-PIN in the response is the cpu,pin of the Entry Manager that made the queue entry.

- **MAX-DATA-SIZE** is the maximum number of bytes to be returned.

- **PAD-CHAR** is the character to be used for padding the reply buffer when the actual data length is less than MAX-DATA-SIZE.

- **RETN-CODE** is the return code. The server returns a code in this field to indicate one of the following entries.

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OK</td>
</tr>
<tr>
<td>4001</td>
<td>W-EOF</td>
</tr>
<tr>
<td>4051</td>
<td>E-MUST-BE-YN</td>
</tr>
<tr>
<td>4052</td>
<td>E-RESERVED-MUST-BE-N</td>
</tr>
<tr>
<td>4078</td>
<td>E-INVALID-PRIORITY</td>
</tr>
<tr>
<td>4087</td>
<td>E-INVALID-MAX-DATASIZE</td>
</tr>
<tr>
<td>4089</td>
<td>W-DATA-TRUNCATED</td>
</tr>
<tr>
<td>4924</td>
<td>E-ERR-QUEUE-FILE</td>
</tr>
<tr>
<td>4990</td>
<td>E-IO-TIMEOUT</td>
</tr>
</tbody>
</table>

- **RETN-CODE-DETAIL** is an error number returned by a subsystem other than the queue management facility or is a further qualification of an error detected by the server.

- **DATA-LEN** is the actual length of the data field before padding the reply buffer to MAX-DATA-SIZE bytes in length.

- The format of the data in the queue entry returned by READQ should be defined by the application. This data would immediately follow the DATA-LEN field and be MAX-DATA-SIZE bytes. The maximum size for the data portion of the response is constrained by the record size defined for the Queue file at configuration time. Refer to the TRANSFER Delivery System Management and Administration Guide.

  A sample format for the data portion would be:

  02 data-field PIC X(100).

**READQ OPERATION.** READQ reads an entry from the named queue. The request returns either an entry from the queue or an error. The error could be an indication there was nothing on the queue.

The response will always contain MAX-DATA-SIZE bytes of data immediately following the DATA-LEN field. If the actual length of the data for the queue entry is less than MAX-DATA-SIZE, or if
Queue Management
READQ

an error occurs during processing, PAD-CHAR will be used to fill in the field.

The entry returned will be the oldest entry from among those with the highest priority.
WAITQ (UOW Code 505)

WAITQ waits for an entry on a queue. This operation does not require a TMF transaction.

```
DEF waitq-uow.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code PIC 9(4) COMP VALUE 505.
  02 flags.
    03 any-queue-name TYPE BOOLEAN VALUE "N".
    03 reserved-1 TYPE BOOLEAN VALUE "N".
    03 reserved-2 TYPE BOOLEAN VALUE "N".
    03 reserved-3 TYPE BOOLEAN VALUE "N".
    03 reserved-4 TYPE BOOLEAN VALUE "N".
    03 reserved-5 TYPE BOOLEAN VALUE "N".
    03 reserved-6 TYPE BOOLEAN VALUE "N".
    03 reserved-7 TYPE BOOLEAN VALUE "N".
  02 queue-name PIC X(32) VALUE SPACES.
  02 wait-priority PIC 9(3) COMP VALUE 0.
  02 wait-timeout PIC S9(4) COMP VALUE -1.
END.

DEF waitq-rsp.
  02 hdr.
    03 self-ident PIC AA VALUE "UW".
    03 uow-code PIC 9(4) COMP VALUE 505.
  02 retn-code PIC S9(4) COMP.
  02 retn-code-detail PIC S9(4) COMP.
  02 queue-name PIC X(32) VALUE SPACES.
END.
```

WAITQ FIELDS. The fields defined in this UOW are:

- HDR is the UOW header. The UOW-CODE value is 505.

- FLAGS lets you qualify the wait operation. Fields within FLAGS are as follows:

  ANY-QUEUE-NAME specifies whether or not the wait is restricted to a specific queue.
Queue Management

WAITQ

Y = Wait for an entry on any queue.

N = Wait for an entry on the queue entered in the QUEUE-NAME field of the request.

RESERVED-1 through RESERVED-7 are reserved for future use; these fields must be set to N.

- QUEUE-NAME identifies the specific queue for the wait request.
- WAIT-PRIORITY determines which waiting consumer is responded to first. The field can contain an unsigned value ranging from 0 (lowest priority) through 199 (highest priority). Higher numbers will be processed before lower numbers.
- WAIT-TIMEOUT sets an upperbound, in seconds, on the amount of time the requesting application is willing to wait for an entry.
  A value of -1 is an unlimited wait.
- RETN-CODE is the return code. The server returns a code in this field to indicate one of the following entries:

  
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OK</td>
</tr>
<tr>
<td>4078</td>
<td>E-INVALID-PRIORITY</td>
</tr>
<tr>
<td>4924</td>
<td>E-ERR-QUEUE-FILE</td>
</tr>
<tr>
<td>4990</td>
<td>E-IO-TIMEOUT</td>
</tr>
<tr>
<td>6002</td>
<td>E-TOO-MANY-WAITERS</td>
</tr>
<tr>
<td>6003</td>
<td>E-INVALID-WAITPRIORITY</td>
</tr>
<tr>
<td>6004</td>
<td>E-INVALID-WAITTIMEOUT</td>
</tr>
<tr>
<td>6005</td>
<td>W-WAIT-TIMEOUT</td>
</tr>
</tbody>
</table>

- RETN-CODE-DETAIL is an error number returned by a subsystem other than the queue management facility or is a further qualification of an error detected by the server.
- QUEUE-NAME in the response is the name of the queue for which a new queue entry has arrived.

WAITQ OPERATION. WAITQ responds to a WAITQ request for a particular queue by reading that queue. If the queue has an entry, WAITQ issues a response; if the queue is empty, WAITQ waits for an entry. When a queue entry is made and WAITQ issues a response to a consumer, the consumer is generally expected to start a transaction and then issue a DEQ request.

If the queue entry is made and then the transaction is aborted, the queue entry might disappear. It is possible, therefore, that the DEQ will return with queue empty even immediately after the WAITQ returns indicating that the queue is not empty. In this case, the consumer should end the transaction and call WAITQ.
again. If the DEQ returns with an E-IO-TIMEOUT error, the application should delay for a short while and retry. The entry might still be locked because the supplier has not yet ended the transaction.

The Wait Manager has no way of knowing if a consumer has canceled its request; to ensure that an available waiting consumer is notified of a new arrival on a queue, the Wait Manager replies to all consumers waiting on the newly added queue entry.

Applications that expect to cancel the WAITQ request based on a timeout should use the WAIT-TIMEOUT feature of this UOW. This will allow the Wait Manager to reclaim space in internal tables as soon as the waiting consumer is not interested in being notified. Another consumer process will then be able to access the Wait Manager.

The following rules apply to WAITQ:

1. Only one WAITQ UOW can appear in a request. If you specify more than one UOW for the Wait Manager, the PW-REPLY-CODE will be set to RQST-ERR (3) and the IPC-RETN-CODE field will be set to RQST-TOO_LONG (5).

2. The WAITQ UOW must be the only UOW in the request. The Wait Manager will not process any portion of the message buffer after the WAITQ UOW.
Queue Management

STARTING THE SERVER PROCESSES

The Entry Manager and the Wait Manager are separate server programs. Both servers interpret the startup message as follows:

- default subvolume - ignored
- IN file name - ignored
- OUT file name - used to write serious error messages; these messages usually indicate a problem in the data base or the operational environment, not in the client
- parameter string - ignored

Table 9-2 lists the parameters that the servers accept at startup time. Unless specified otherwise, the parameters apply to both servers.

Table 9-2. Queue Manager Server Parameters at Startup Time

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEBUGLOGFORMAT</td>
<td>FALSE</td>
<td>If FALSE: the debugging log will contain binary characters suitable for writing to an entry-sequenced disc file. If TRUE: the debugging log will be formatted in ASCII characters suitable for writing to a terminal or printer.</td>
</tr>
<tr>
<td>DEBUGLOGLEVEL</td>
<td>3</td>
<td>If the PW-REPLY-CODE field of any response is greater than or equal to this value, the message and reply will be written to the debugging log. Specifying a value of zero causes all messages and replies to be logged.</td>
</tr>
<tr>
<td>Parameter Name</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DEBUGLOGRECSPEROPEN</td>
<td>20</td>
<td>After this many IPCs and their replies are written to the debug log, the log is closed and reopened.</td>
</tr>
<tr>
<td>HANGAROUND</td>
<td>FALSE</td>
<td>If FALSE: the server stops after all requesters have closed it. If TRUE: the server never stops automatically.</td>
</tr>
<tr>
<td>CAUTION</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>When requesters and servers are both running in the PATHWAY environment, the HANGAROUND parameter for the PATHWAY server must be set to FALSE.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When a requester is running outside the PATHWAY environment and accessing a PATHWAY server, that PATHWAY server must be in its own server class so the HANGAROUND parameter can be set to TRUE; with the TRUE setting, the server remains available after it has been closed by the last opener.</td>
</tr>
<tr>
<td>IOTIMEOUT</td>
<td>20</td>
<td>Longest time, in seconds, to wait for I/O requests that are blocked due to other transactions.</td>
</tr>
<tr>
<td>MAXLINKS</td>
<td>8</td>
<td>Number of simultaneous openers of a server process.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Entry Manager: The actual limit is related to MAXLINKS and the values for MAXREQUEST and MAXREPLY.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Wait Manager: The number must not be more than 250; a larger value will cause a diagnostic to be issued and the server to ABEND.</td>
</tr>
</tbody>
</table>
Table 9-2. Queue Manager Server Parameters at Startup Time (Continued)

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAXREPLY</td>
<td>2048</td>
<td>Size (in bytes) of largest reply a server can generate. This might affect the maximum number of simultaneous openers due to memory limitations.</td>
</tr>
<tr>
<td></td>
<td>Entry Manager only</td>
<td></td>
</tr>
<tr>
<td>MAXREQUEST</td>
<td>2048</td>
<td>Size (in bytes) of largest request a server will accept. This might affect the maximum number of simultaneous openers due to memory limitations.</td>
</tr>
<tr>
<td></td>
<td>Entry Manager only</td>
<td></td>
</tr>
<tr>
<td>MAXWAITERS</td>
<td>MAXLINKS</td>
<td>Number of simultaneous WAITQs that can be outstanding at any one time. The number must not be more than 250; a larger value will cause a diagnostic to be issued and the server to ABEND.</td>
</tr>
<tr>
<td></td>
<td>Wait Manager only</td>
<td></td>
</tr>
<tr>
<td>RESERVEDLINKS</td>
<td>1</td>
<td>Number of openers (set by MAXLINKS) that must be Entry Manager programs rather than application components. The number of application components that can open the Wait Manager is MAXLINKS minus RESERVEDLINKS.</td>
</tr>
<tr>
<td></td>
<td>Wait Manager only</td>
<td></td>
</tr>
<tr>
<td>Parameter Name</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>RESERVELCBS</td>
<td>0</td>
<td>Number of LCBs that a Wait Manager reserves for its own use. Use of this parameter allows the message about outstanding requests on a process to be removed. If your Wait Manager will typically have a large number of waiters outstanding, this probably should be set to a number that corresponds to the expected number of simultaneous waiters.</td>
</tr>
<tr>
<td>WAKEUPINTERVAL</td>
<td>60 seconds</td>
<td>Amount of time the Wait Manager will sit idle waiting on $RECEIVE. After this much idle time has passed, the Wait Manager will check the Queue file to make sure there are no entries to be processed (entries could be there if a transaction with a DEQ aborts, for example). If there are no outstanding requests, the Wait Manager will wake up every WAKEUPINTERVAL and perform a check on a single opener. One opener is checked every WAKEUPINTERVAL until all openers have been checked. (MAXLINKS x WAKEUPINTERVAL) is the amount of idle time that will pass before all checks have been made and the Wait Manager starts over from the beginning.</td>
</tr>
</tbody>
</table>
Queue Management

Logical File Assignments

The servers accept parameters in the ASSIGN command at initialization time for several logical files. These logical files allow some control over the physical files used by the queue management servers.

ASSIGN command parameters are listed and described in Table 9-3. For additional information, refer to the TRANSFER Delivery System Management and Administration Guide.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEBUGLOG</td>
<td>None</td>
<td>Specifies the file to which a debugging log is written. The log contains a copy of some or all of the IPCs received and replied to by the servers. Control over log entries is provided by the parameters listed in Table 9-2 and by a flag field in the IPC header. If this parameter is omitted, a debugging log is not written. If this parameter refers to a disc file that does not exist, an entry-sequenced file will be created. This parameter cannot refer to an EDIT file.</td>
</tr>
<tr>
<td>QUEUEFILE</td>
<td>QUEUE (in the volume and subvolume on which the server program resides)</td>
<td>Specifies the file that contains all queue information and that is opened and used by both servers. The file must exist and must be audited; otherwise, an error message is reported and the server ABENDs.</td>
</tr>
</tbody>
</table>
Table 9-3. Queue Manager ASSIGN Command Parameters at Startup Time (Continued)

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WAITMANAGER</td>
<td>None</td>
<td>Specifies the process name of the Wait Manager that is handling wait requests for new entries on the Queue file. This parameter is required by all Entry Managers operating on a single Queue file. If a value is not supplied, the Entry Manager reports an error and ABENDs.</td>
</tr>
</tbody>
</table>

Server Program Files

The Entry Manager and Wait Manager are each in one program file. The Entry Manager server is named EMSERV. The server handles requests one at a time even when there are multiple openers. The Wait Manager server is named WMSERV. The server handles multiple requests concurrently.

The servers allow access to queue data only if the requesting process has read and write access to the Queue file. Both servers check security when they receive an OPEN request from the requester. If the requester does not have access privilege, the servers deny access and issue a security violation.

A security violation (error 48) implies that the opening process does not have a user-id falling within the access security of the Queue file. Either the user-id of the requesting process or the security mask on the file must be changed before the process can successfully open the server.

QUEUE MANAGEMENT INITIALIZATION ERROR MESSAGES

Error messages that can be issued during initialization are listed in Table 9-4. Errors reported by the queue management UOWs appear in Appendix A.
Table 9-4. Queue Management Initialization Error Messages

<table>
<thead>
<tr>
<th>Message</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bad version number on file &lt;filename&gt;</td>
<td>The version number in the control record of the specified file was invalid for the running version of the server. The file must be converted to the current compatible version.</td>
</tr>
<tr>
<td>Defined data structure queue-file-control inconsistent</td>
<td>The data within the structure is inconsistent with the length of the structure itself. This will cause the particular server to ABEND. There might be a problem with the compilation procedure used to build the server. See your Tandem representative.</td>
</tr>
<tr>
<td>Error &lt;error&gt; obtaining maximum record length of file &lt;filename&gt;</td>
<td>An error occurred when the server attempted to obtain the maximum record length of the specified file. The &lt;error&gt; value is the file system error number to see for additional information.</td>
</tr>
<tr>
<td>Error &lt;error&gt; obtaining information for file &lt;filename&gt;</td>
<td>An error occurred when the server attempted to obtain information about the specified file. The &lt;error&gt; value is the file system error number to see for additional information.</td>
</tr>
<tr>
<td>Error &lt;error&gt; opening file &lt;filename&gt;</td>
<td>The attempt to open the specified file failed. The &lt;error&gt; value is the file system error number to see for additional information.</td>
</tr>
</tbody>
</table>
Table 9-4. Queue Management Initialization Error Messages (Continued)

<table>
<thead>
<tr>
<th>Message</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error <code>&lt;error&gt;</code> while setting lock mode of file <code>&lt;filename&gt;</code></td>
<td>The specified file system error occurred while trying to set the locking mode of the specified file. The <code>&lt;error&gt;</code> value is the file system error number to see for additional information.</td>
</tr>
<tr>
<td>File code incorrect for file <code>&lt;filename&gt;</code></td>
<td>The file code for the specified file is incorrect. The file must be recreated with the correct file code.</td>
</tr>
<tr>
<td>File <code>&lt;filename&gt;</code> must be audited</td>
<td>The specified file is not a TMF audited file. The file must be specified as audited.</td>
</tr>
<tr>
<td>Insufficient memory for required Wait Manager tables</td>
<td>The Wait Manager could not obtain sufficient memory for its tables. The parameters defining the number of openers andwaiters supported by the Wait Manager must be reduced to conform to memory restrictions.</td>
</tr>
<tr>
<td>Invalid event value in Wait Manager</td>
<td>The event value that designates Wait Manager action was invalid; the server will ABEND. Starting the server again might solve the problem; if not, the problem is in the server code.</td>
</tr>
</tbody>
</table>
Table 9-4. Queue Management Initialization
Error Messages (Continued)

<table>
<thead>
<tr>
<th>Message</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>KEYPOSITION error &lt;error&gt; on file &lt;filename&gt;</td>
<td>The specified file system error occurred when a KEYPOSITION was attempted on the specified file. The &lt;error&gt; value is the file system error number to see for additional information.</td>
</tr>
<tr>
<td>No control record found in file &lt;filename&gt;</td>
<td>The specified file contained records, but no control record. The file must be recreated so that a control record is written.</td>
</tr>
<tr>
<td>MAXLINKS value is too large</td>
<td>The MAXLINKS parameter has a value that is greater than the defined maximum. The value must be reduced and the server started again.</td>
</tr>
<tr>
<td>MAXWAITERS value is too large</td>
<td>The MAXWAITERS parameter has a value that is greater than the defined maximum. The value must be reduced and the server started again.</td>
</tr>
<tr>
<td>READ error &lt;error&gt; on file &lt;filename&gt;</td>
<td>The specified file system error occurred when a READ was attempted on the specified file. The &lt;error&gt; value is the file system error number to see for additional information.</td>
</tr>
</tbody>
</table>
Table 9-4. Queue Management Initialization
Error Messages (Continued)

<table>
<thead>
<tr>
<th>Message</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unable to reserve lcb's</td>
<td>The server was unable to reserve the required lcb's. The number of RESERVELCBS must be reduced and the server started again.</td>
</tr>
<tr>
<td>Unexpected error &lt;error&gt; on file &lt;filename&gt;</td>
<td>An unexpected error is returned from the file system. The specified &lt;error&gt; indicates the problem.</td>
</tr>
<tr>
<td>WRITE error &lt;error&gt; on file &lt;filename&gt;</td>
<td>The specified file system error occurred when a WRITE was attempted on the specified file. The &lt;error&gt; value is the file system error number to see for additional information.</td>
</tr>
</tbody>
</table>
APPENDIX A

ERROR CODES

Error codes can be returned to a TRANSFER application process in the following fields:

- **PW-REPLY-CODE**
- **IPC-RETN-CODE**
- **IPC-RETN-CODE-DETAIL**
- **RETN-CODE**
- **RETN-CODE-DETAIL**
- **MBR-RETN-CODE**
- **ELEM-RETN-CODE**
- **DELIV-ERR**

in the interprocess communication (IPC) header

in the UOW header

in the ADD-MEMBER and DELETE-MEMBER UOW responses

in the ALTER-PROFILE-ELEM and GET-PROFILE-ELEM UOW responses

in the GET-RECIP-REC UOW response

ERROR CODE FIELDS

The following paragraphs give a general discussion of the fields that contain error codes. All error codes are listed and explained in this appendix.

PW-REPLY-CODE Errors

A value of zero in the PW-REPLY-CODE field of the IPC header indicates that all UOWs in the request were processed successfully. Any other value denotes an error. Additional information about these errors appears in Section 4.
Error Codes

CAUTION

A value of 2 or greater in the PW-REPLY-CODE field requires a transaction abort and full transaction backout by TMF.

IPC-RETN-CODE Errors

A value of zero in the IPC-RETN-CODE field of the IPC header indicates that TRANSFER detected no errors in the header; warning indications or errors, however, might have been detected in individual UOWs. Any other value in this field denotes an error in the header, in a file access operation, or in some other area that is not specific to any particular UOW in the request.

IPC-RETN-CODE-DETAIL Errors

A value other than zero in the IPC-RETN-CODE-DETAIL field of the IPC header indicates that although the data in the IPC header was correct, an error was detected by the GUARDIAN operating system. The meaning of this error also depends upon the content of the IPC-RETN-CODE field. For information about these particular errors, refer to the GUARDIAN Operating System Programming Manual.

RETN-CODE Errors

A value of zero in the RETN-CODE field of the UOW header indicates that no errors occurred during the processing of the particular UOW. Any other value in this field is one of the following:

- an error code - a positive value ranging from 4000 through 5999, denoting that the UOW could not be processed because of an error.

- a warning code - a negative value ranging from -5999 through -4000, denoting that the UOW contained an anomaly; but in spite of the anomaly, the UOW was processed successfully.

Error code text for these errors begins with E and warning code text begins with W.
The range within which the error or warning code number falls indicates which process detected the anomaly, as follows:

<table>
<thead>
<tr>
<th>Positive/Negative Range</th>
<th>Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>4000 through 4399</td>
<td>TRANSFER interactive server (TISERV)</td>
</tr>
<tr>
<td>4600 through 4699</td>
<td>TRANSFER asynchronous processors</td>
</tr>
<tr>
<td>4900 through 4999</td>
<td>GUARDIAN file system</td>
</tr>
<tr>
<td>5600 through 5799</td>
<td>TRANSFER name server</td>
</tr>
<tr>
<td>6000 through 6099</td>
<td>TRANSFER queue manager</td>
</tr>
</tbody>
</table>

TRANSFER issues RETN-CODE errors in the range 4000 through 6099. This appendix lists only those codes returned by UOWs in the RETN-CODE field, or written by TAREQs in the DELIV-ERR field of recipient records. Other codes in this range can be generated by the T/Mail application and the ADMIN client; the messages associated with these codes appear in the TRANSFER/MAIL User's Guide and the TRANSFER Delivery System Management and Administration Guide.

All UOW code numbers that are passed to TRANSFER by your application and that are not explicitly listed in this appendix return one of these errors to the RETN-CODE field:

- **E-INVALID-UOW-HDR**
  This is a request error. The UOW code is undefined in the system.

- **E-RESTRICTED-OPERATION**
  This is a UOW error. The UOW is reserved for use by TAREQs.

- **E-UOW-NOT-IMPLEMENTED**
  This is a UOW error. The code is reserved for a UOW that is not yet implemented.

**RETN-CODE-DETAIL Errors**

Certain errors reported in the RETN-CODE field also cause additional related error codes to appear in the RETN-CODE-DETAIL field. These additional codes can reflect:

- Errors detected by the GUARDIAN operating system. For information about such errors, refer to the GUARDIAN Operating System Programming Manual.

- Errors detected by the TRANSFER name server. These errors are reported almost identically to the way in which file errors detected by the GUARDIAN operating system are reported. The messages that accompany the name server errors have approximately the same meaning as similar messages generated by the operating system.
Error Codes

• Errors detected by TRANSFER indicating an improper entry in an absolute DATE-TIME subfield, as follows:

<table>
<thead>
<tr>
<th>Error Value Returns in</th>
<th>DATE-TIME Subfield Containing Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>YEAR</td>
</tr>
<tr>
<td>2</td>
<td>MONTH</td>
</tr>
<tr>
<td>3</td>
<td>DAY-OF-MONTH</td>
</tr>
<tr>
<td>4</td>
<td>HOUR</td>
</tr>
<tr>
<td>5</td>
<td>MINUTE</td>
</tr>
<tr>
<td>6</td>
<td>SECOND</td>
</tr>
</tbody>
</table>

MBR-RETN-CODE Errors

The MBR-RETN-CODE field of the ADD-MEMBER and DELETE-MEMBER UOWs can contain error codes numbered from 5650 through 5674.

ELEM-RETN-CODE Errors

Error codes can sometimes be returned in the responses to the ALTER-PROFILE-ELEM and GET-PROFILE-ELEM UOWs. These codes are listed in the description of the GET-PROFILE-ELEM UOW.

DELIV-ERR Errors

Error codes can be returned by TAREQs in the response to the GET-RECIP-REC UOW. These error codes are numbered from 4600 through 4609.

GUARDIAN FILE CODES

GUARDIAN file codes can be returned in some error code fields. These codes indicate the TRANSFER file on which the error occurred. File codes are listed and identified in Table A-1.
### Table A-1. GUARDIAN File Codes

<table>
<thead>
<tr>
<th>File Code</th>
<th>File Name</th>
<th>File Code</th>
<th>File Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>250</td>
<td>Profile</td>
<td>259</td>
<td>Net</td>
</tr>
<tr>
<td>251</td>
<td>Session</td>
<td>260</td>
<td>Inverted Folder</td>
</tr>
<tr>
<td>252</td>
<td>Item Descriptor</td>
<td>261</td>
<td>Restart</td>
</tr>
<tr>
<td>253</td>
<td>Recipient</td>
<td>262</td>
<td>Name</td>
</tr>
<tr>
<td>254</td>
<td>Folder</td>
<td>263</td>
<td>DIN</td>
</tr>
<tr>
<td>255</td>
<td>Item Data</td>
<td>264</td>
<td>Alias</td>
</tr>
<tr>
<td>256</td>
<td>Distribution List</td>
<td></td>
<td>Trace</td>
</tr>
<tr>
<td>257</td>
<td>Ready</td>
<td>265</td>
<td></td>
</tr>
<tr>
<td>258</td>
<td>Time</td>
<td>266</td>
<td>Queue File</td>
</tr>
</tbody>
</table>

### SUMMARY OF ERROR CODES

Errors reported by TRANSFER are listed in Table A-2. Error codes appear in ascending numeric order.

The first column shows the error number associated with each error and the field in which the error is reported.

The second column shows the mnemonic text that describes the error and the specific meaning of the error.

Error code text begins with E; warning code text begins with W.

The minus signs that precede the numeric codes for warnings are omitted.

Errors listed in Table A-2 are also listed in Table A-3 for cross-reference. Error messages in Table A-3 appear in alphabetic order.
## Error Codes

### Table A-2. Error Codes

<table>
<thead>
<tr>
<th>Error No./ (Field)</th>
<th>Text/ Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (PW-REPLY-CODE)</td>
<td>UOWS-WITH-WARNING</td>
</tr>
<tr>
<td></td>
<td>TRANSFER encountered warning indications in one or more UOWs. If STOP-ON-WARNING was not specified in the RQST-CODE field, then all UOWs in the request have a corresponding response UOW in the reply. If STOP-ON-WARNING was specified, only those UOWs preceding and including the first with a warning indication have corresponding response UOWs.</td>
</tr>
<tr>
<td>1 (IPC-RETN-CODE)</td>
<td>INVALID-VERSION-CODE</td>
</tr>
<tr>
<td></td>
<td>The request contained an IPC version code that could not be recognized by TRANSFER.</td>
</tr>
<tr>
<td>2 (PW-REPLY-CODE)</td>
<td>UOWS-WITH-ERR</td>
</tr>
<tr>
<td></td>
<td>TRANSFER detected errors in one or more UOWs. If neither STOP-ON-ERR nor STOP-ON-WARNING was specified in the RQST-CODE field, then all UOWs in the request have a corresponding response UOW; otherwise, only those UOWs preceding and including the first with an error have corresponding response UOWs. (The UOWs preceding the first with an error might return warning indications if STOP-ON-ERR is specified in the request.)</td>
</tr>
<tr>
<td></td>
<td>CAUTION</td>
</tr>
<tr>
<td></td>
<td>If your application receives a value of 2 in the PW-REPLY-CODE field, full transaction backout with TMF is recommended. If the transaction is not backed out, consistency in the data base cannot be guaranteed.</td>
</tr>
</tbody>
</table>
### Error Codes

<table>
<thead>
<tr>
<th>Error No./ (Field)</th>
<th>Text/ Meaning</th>
</tr>
</thead>
</table>
| 2 (IPC-RETN-CODE)  | INVALID-SESSION-ID  
The request contained a session ID that could not be recognized by TRANSFER. Perhaps the IPC referenced a session that no longer existed. |
| 3 (PW-REPLY-CODE)  | RQST-ERR      
A request error occurred. This type of error indicates that something was wrong in the data in the IPC header, and that the error probably does not apply to any particular UOW. In certain cases, however, it might involve an individual UOW, for example, one with an invalid UOW header. When this error occurs, the number of response UOWs might be less than the number of UOWs in the request. For further information, your application should examine the IPC-RETN-CODE field. |
| 3 (IPC-RETN-CODE)  | SERVICE-DENIED  
Reserved for future use by Tandem Computers. |

**CAUTION**

If your application receives a value of 3 in the PW-REPLY-CODE field, full transaction backout with TMF is recommended. If the transaction is not backed out, consistency in the TRANSFER data base cannot be guaranteed.
Table A-2. Error Codes (Continued)

<table>
<thead>
<tr>
<th>Error No./ (Field)</th>
<th>Text/ Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 (IPC-RETN-CODE)</td>
<td>INVALID-UOW-HDR</td>
</tr>
<tr>
<td></td>
<td>Your request contained an invalid UOW header. This error occurs when the previous UOW header specified the wrong size, or when you omitted the characters UW from the UOW header, or when the UOW code in the UOW header is undefined for the server class.</td>
</tr>
<tr>
<td>5 (IPC-RETN-CODE)</td>
<td>RQST-TOO-LONG</td>
</tr>
<tr>
<td></td>
<td>The request was too long to be handled by TISERV. (The TISERV parameter MAXREQUEST specifies the size of the largest request that TISERV can process.)</td>
</tr>
<tr>
<td>6 (IPC-RETN-CODE)</td>
<td>REPLY-TOO-LONG</td>
</tr>
<tr>
<td></td>
<td>The reply from the server was too long to fit into the allocated reply buffer.</td>
</tr>
<tr>
<td>7 (IPC-RETN-CODE)</td>
<td>RQST-TOO-SHORT</td>
</tr>
<tr>
<td></td>
<td>The request was too short; it contained fewer UOWs than the number specified in the UOWS-TO-PROCESS field.</td>
</tr>
<tr>
<td>8 (IPC-RETN-CODE)</td>
<td>INVALID-REQUEST-CODE</td>
</tr>
<tr>
<td></td>
<td>The RQST-CODE field contained an invalid entry.</td>
</tr>
</tbody>
</table>
## Error Codes

### Table A-2. Error Codes (Continued)

<table>
<thead>
<tr>
<th>Error No./ (Field)</th>
<th>Text/ Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>4001 (RETN-CODE)</td>
<td>W-EOF</td>
</tr>
<tr>
<td></td>
<td>While reading information from its data base, TRANSFER encountered an end-of-file indicator. This error occurs with such UOWs as GET-ITEM-REC, GET-RECIP-REC, READ-NEXT-MEMBER, READ-NEXT-NAME, or READ-PROFILE-REC. Since TRANSFER has returned to your program all available data in the file, your process should not call the UOW again for more data.</td>
</tr>
<tr>
<td>4003 (RETN-CODE)</td>
<td>E-UOW-NOT-IMPLEMENTED</td>
</tr>
<tr>
<td></td>
<td>Your process issued a UOW that is not implemented in TRANSFER.</td>
</tr>
<tr>
<td>4005 (RETN-CODE)</td>
<td>E-CONCURRENT-SESSION W-CONCURRENT-SESSION</td>
</tr>
<tr>
<td></td>
<td>Your application attempted to start a session for a correspondent (with the START-SESSION UOW) while another session was already in progress for the same correspondent. This can be either an error or a warning message, depending on the configuration of the depot involved.</td>
</tr>
<tr>
<td>4007 (RETN-CODE)</td>
<td>E-LOGON-DISALLOWED</td>
</tr>
<tr>
<td></td>
<td>A correspondent attempted to log on, but a system administrator has set the LOGON-DISALLOWED field in the correspondent's profile record to Y, preventing anyone from logging on under that correspondent's name.</td>
</tr>
<tr>
<td>4009 (RETN-CODE)</td>
<td>E-WRITES-DISALLOWED</td>
</tr>
<tr>
<td></td>
<td>Your process attempted a write operation, but writing has been explicitly disallowed.</td>
</tr>
<tr>
<td>Error No./ (Field)</td>
<td>Text/ Meaning</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>4010</td>
<td>E-BAD-TRANSACTION</td>
</tr>
<tr>
<td>(IPC-RETN-CODE or RETN-CODE)</td>
<td>Your process issued a UOW without first obtaining the required transaction ID; or the current transaction could not be used, probably because some component of the system or network failed. See IPC-RETN-CODE-DETAIL or RETN-CODE-DETAIL for the file system error returned by the GUARDIAN operating system.</td>
</tr>
<tr>
<td>4013</td>
<td>E-NO-DEPOT-FOR-CORR</td>
</tr>
<tr>
<td>(RETN-CODE)</td>
<td>Your process attempted to start a session for a correspondent whose name is defined in the TRANSFER name directory, but for whom no profile record currently exists.</td>
</tr>
<tr>
<td>4015</td>
<td>E-INVALID-PASSWORD</td>
</tr>
<tr>
<td>(RETN-CODE)</td>
<td>Your process attempted to start a session for a correspondent who entered an invalid password.</td>
</tr>
<tr>
<td>4017</td>
<td>E-RESTRICTED-OPERATION</td>
</tr>
<tr>
<td>(RETN-CODE)</td>
<td>Your process issued a UOW that should only be issued by TRANSFER.</td>
</tr>
<tr>
<td>4019</td>
<td>E-ALREADY-IN-SESSION</td>
</tr>
<tr>
<td>(RETN-CODE)</td>
<td>Your process issued a START-SESSION UOW without first resetting the session ID in the IPC header.</td>
</tr>
<tr>
<td>Error No./ (Field)</td>
<td>Text/ Meaning</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>4021 (RETN-CODE)</td>
<td>E-INVALID-ZONE-OFFSET</td>
</tr>
<tr>
<td></td>
<td>The CURR-LOCAL-TIME specified in the START-SESSION UOW differs from the local time of the TISERV server (to which the UOW was issued) by more than 24 hours. This error normally indicates that your client specified the local time incorrectly.</td>
</tr>
<tr>
<td>4035 (RETN-CODE)</td>
<td>E-ITEM-NOT-FOUND</td>
</tr>
<tr>
<td></td>
<td>Your process referenced an item that was not:</td>
</tr>
<tr>
<td></td>
<td>• saved in any of your folders,</td>
</tr>
<tr>
<td></td>
<td>• created by you in the current session, or</td>
</tr>
<tr>
<td></td>
<td>• returned by TRANSFER in response to a GET-ITEM-COMPNT-A01 UOW in the current session.</td>
</tr>
<tr>
<td>4036 (RETN-CODE)</td>
<td>E-ITEM-TOO-COMPLEX</td>
</tr>
<tr>
<td></td>
<td>Your process issued an ATTACH-COMPNT or SUBMIT-PKG UOW for an item that involved too many levels of nesting. Restructure the item to include fewer levels, and retry the request.</td>
</tr>
<tr>
<td>4037 (RETN-CODE)</td>
<td>E-PARENT-NOT-FOUND</td>
</tr>
<tr>
<td></td>
<td>Your process attempted to attach or detach an item as a component of a parent item that was not:</td>
</tr>
<tr>
<td></td>
<td>• saved in any of your folders,</td>
</tr>
<tr>
<td></td>
<td>• created by you in the current session, or</td>
</tr>
<tr>
<td></td>
<td>• returned by TRANSFER in response to a GET-ITEM-COMPNT-A01 UOW in current session.</td>
</tr>
</tbody>
</table>
## Error Codes

### Table A-2. Error Codes (Continued)

<table>
<thead>
<tr>
<th>Error No./ (Field)</th>
<th>Text/ Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>4038</td>
<td>E-COMPNT-CYCLE</td>
</tr>
<tr>
<td>(RETN-CODE)</td>
<td>Your process attempted to attach one item as a component of another (with the ATTACH-COMPNT UOW), but the first item was already defined as an ancestor of the second.</td>
</tr>
<tr>
<td>4039</td>
<td>E-COMPNT-NOT-FOUND</td>
</tr>
</tbody>
</table>
| (RETN-CODE)        | Your process attempted to attach or detach one item as a component of another, but the component item was not:  
|                    | • saved in any of your folders, |
|                    | • created by you in the current session, or |
|                    | • returned by TRANSFER in response to a GET-ITEM-COMPNT-A01 UOW in the current session. |
| 4040               | E-BAD-ITEM-DESCR |
| (RETN-CODE)        | Your process attempted to attach a component (ATTACH-COMPNT-A01 UOW), detach a component (DETACH-COMPNT UOW), retrieve a component (GET-ITEM-COMPNT-A01 UOW), or post a package (SUBMIT-PKG UOW); but TRANSFER detected an internal inconsistency for the referenced item in its descriptor. |
| 4041               | E-ITEM-UNALTERABLE |
| (RETN-CODE)        | Your process attempted to alter an item that was part of a package already submitted for delivery, or to change the recipient list for such a package. |
Table A-2. Error Codes (Continued)

<table>
<thead>
<tr>
<th>Error No./ (Field)</th>
<th>Text/ Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>4042 (RETN-CODE)</td>
<td>E-ITEM-NOT-PKG-HDR</td>
</tr>
<tr>
<td></td>
<td>Your process tried to perform an operation with a package, but referenced an item that was not a package header. This error might occur while acknowledging receipt of a package, reading or altering its recipient list, altering the item descriptor, or submitting or canceling the package.</td>
</tr>
<tr>
<td>4043 (RETN-CODE)</td>
<td>E-PREVIOUSLY-SUBMITTED</td>
</tr>
<tr>
<td></td>
<td>Your process tried to post (SUBMIT-PKG UOW) a package that was already posted.</td>
</tr>
<tr>
<td>4044 (RETN-CODE)</td>
<td>E-PREVIOUSLY-CANCELED</td>
</tr>
<tr>
<td></td>
<td>Your process tried to cancel (CANCEL-PKG UOW) a package that was already canceled.</td>
</tr>
<tr>
<td>4045 (RETN-CODE)</td>
<td>E-TSCHED-UNAVAIL</td>
</tr>
<tr>
<td></td>
<td>Your process requested a function that required asynchronous services (posting or canceling a package, or acknowledging its receipt), but these services were not available. If this error occurs when your process is saving an item with the SAVE-ITEM-B00 or SAVE-ITEM-BY-KEY UOW, you must either wait for TSCHED to become available or save the item without an unsave date and time specified.</td>
</tr>
<tr>
<td>4046 (RETN-CODE)</td>
<td>E-INVALID-REC-TYPE</td>
</tr>
<tr>
<td></td>
<td>When reading, writing, or deleting a profile or item data record, your process specified a record type that was either reserved for use by TRANSFER only or was outside the range 1 through 9999.</td>
</tr>
</tbody>
</table>
### Table A-2. Error Codes (Continued)

<table>
<thead>
<tr>
<th>Error No./ (Field)</th>
<th>Text/ Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>4047 (RETN-CODE)</td>
<td><strong>E-REC-NOT-FOUND</strong>&lt;br&gt;When altering agent selection criteria, deleting a data record from an item, or purging a recipient or correspondent's profile, your process referenced a record that could not be found in the TRANSFER database.</td>
</tr>
<tr>
<td>4049 (RETN-CODE)</td>
<td><strong>E-REC-ALREADY-EXISTS</strong>&lt;br&gt;When adding a record to an item or a recipient to a recipient list, or when saving an item or writing a profile record, your process tried to create a record that already existed.</td>
</tr>
<tr>
<td>4050 (RETN-CODE)</td>
<td><strong>E-UPDATE-MISMATCH</strong>&lt;br&gt;Your process tried to update a record with the ALTER-ITEM-DESCR or WRITE-PROFILE-REC UOW, but the UPDATE-CONTROL field for the record does not match the current update control count maintained by TRANSFER. Either your process returned the wrong record key, or another process updated the record between the time that it was read and the time that it was written.</td>
</tr>
<tr>
<td>4051 (RETN-CODE)</td>
<td><strong>E-MUST-BE-YN</strong>&lt;br&gt;Your process attempted to set a flag in the TRANSFER data base to a value other than Y, N, y, or n.</td>
</tr>
</tbody>
</table>
Table A-2. Error Codes (Continued)

<table>
<thead>
<tr>
<th>Error No./ (Field)</th>
<th>Text/ Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>4052 (RETN-CODE)</td>
<td>E-RESERVED-MUST-BE-N</td>
</tr>
<tr>
<td></td>
<td>Your process tried to set a TRANSFER database flag that is reserved for future use by TRANSFER. Such flags are initialized to N, and must remain at this setting.</td>
</tr>
<tr>
<td>4054 (RETN-CODE)</td>
<td>E-INVALID-AGENT-SEL</td>
</tr>
<tr>
<td></td>
<td>Through the ALTER-AGENT-SELECT UOW, your process entered a value in the AGENT-SEL-LOW or AGENT-SEL-HIGH field that was outside the range of 0 through 9999; or in the ALTER-ITEM-DESCR UOW, you specified an agent selector outside the range of 0 through 9999.</td>
</tr>
<tr>
<td>4055 (RETN-CODE)</td>
<td>E-INVALID-APPLIC-ID</td>
</tr>
<tr>
<td></td>
<td>Through the ALTER-AGENT-SELECT UOW, your process entered a value in the APPLIC-ID-LOW, APPLIC-ID-HIGH, or APPLIC-ID-FOR-LOGON field that was outside the range of 0 through 9999; or, in the START-SESSION UOW, you specified an APPLIC-ID outside the range of 0 through 9999.</td>
</tr>
<tr>
<td>4056 (RETN-CODE)</td>
<td>E-INVALID-ITEM-TYPE</td>
</tr>
<tr>
<td></td>
<td>Your process entered an item type value that was outside the range of 0 through 9999.</td>
</tr>
<tr>
<td>4057 (RETN-CODE)</td>
<td>E-INVALID-REL-POSITION</td>
</tr>
<tr>
<td></td>
<td>While attaching, detaching, or retrieving a component item (with the ATTACH-COMPNT, DETACH-COMPNT, or GET-ITEM-COMPNT UOWs, respectively), your process specified a relative position for the item within the component list that was out of range.</td>
</tr>
</tbody>
</table>
Error Codes

Table A-2. Error Codes (Continued)

<table>
<thead>
<tr>
<th>Error No./ (Field)</th>
<th>Text/ Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>4058 (RETN-CODE)</td>
<td>E-INVALID-REC-SEQ-NUM When altering agent selection criteria (with the ALTER-AGENT-SELECT UOW), when writing a profile record (with the WRITE-PROFILE-REC UOW), or when adding an item data record (with the ADD-ITEM-REC UOW), your process specified a record sequence number that was not within the range of 0 through 9999.</td>
</tr>
<tr>
<td>4061 (RETN-CODE)</td>
<td>W-ITEM-NOT-IN-FOLDER While removing an item from a folder (with the UNSAVE-ITEM UOW), your process referenced an item that was not present in the folder.</td>
</tr>
<tr>
<td>4065 (RETN-CODE)</td>
<td>E-INVALID-RECIP-TYPE In the ADD-RECIP UOW, your process specified a recipient type that was not within the range of 0 through 9999.</td>
</tr>
<tr>
<td>4067 (RETN-CODE)</td>
<td>W-REMOTE-NAME-ACCEPTED Your process specified a correspondent or distribution list supposedly defined at a remote node. At your request, TRANSFER accepted that name without validating that it exists. This warning might occur while adding recipients to a package or members to a distribution list.</td>
</tr>
<tr>
<td>4069 (RETN-CODE)</td>
<td>W-NODE-NAME-UNKNOWN While adding a recipient (with the ADD-RECIP or ADD-MEMBER UOW), your process referenced a network node name that was not currently known in the EXPAND network. This message is a warning, indicating that TRANSFER has accepted the name anyway.</td>
</tr>
</tbody>
</table>
### Table A-2. Error Codes (Continued)

<table>
<thead>
<tr>
<th>Error No./ (Field)</th>
<th>Text/ Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>4071 (RETN-CODE)</td>
<td>E-PAST-DATE-TIME</td>
</tr>
<tr>
<td>4072 (RETN-CODE)</td>
<td>E-UNITS-MUST-BE-DHM</td>
</tr>
<tr>
<td>4073 (RETN-CODE)</td>
<td>E-INVALID-DATE-TIME</td>
</tr>
<tr>
<td>4074 (RETN-CODE)</td>
<td>E-INVALID-REL-TIME-QTY</td>
</tr>
<tr>
<td>4075 (RETN-CODE)</td>
<td>W-TIME-WINDOW-EXTENDED</td>
</tr>
</tbody>
</table>
### Error Codes

#### Table A-2. Error Codes (Continued)

<table>
<thead>
<tr>
<th>Error No./ (Field)</th>
<th>Text/ Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>4077 (RETN-CODE)</td>
<td>E-LIFESPAN-TOO-LONG</td>
</tr>
<tr>
<td></td>
<td>In a request to post a package (SUBMIT-PKG UOW), your process submitted a package whose lifetime exceeded the maximum allowed by the system.</td>
</tr>
<tr>
<td>4078 (RETN-CODE)</td>
<td>E-INVALID-PRIORITY</td>
</tr>
<tr>
<td></td>
<td>Your process specified a delivery priority for a package that was outside the range of 0 through 199.</td>
</tr>
<tr>
<td>4079 (RETN-CODE)</td>
<td>W-PRIORITY-REDUCED</td>
</tr>
<tr>
<td></td>
<td>In a request to post a package (SUBMIT-PKG UOW), your process submitted a package with a priority that was too high, and TRANSFER reduced it to the value defined as the limit for the depot.</td>
</tr>
<tr>
<td>4080 (RETN-CODE)</td>
<td>E-PKG-NOT-RECEIVED</td>
</tr>
<tr>
<td></td>
<td>Your process tried to acknowledge (with the ACK-RECEIPT UOW) delivery of a package that was not received by the logged-on correspondent.</td>
</tr>
<tr>
<td>4081 (RETN-CODE)</td>
<td>W-DELIV-IN-PROGRESS</td>
</tr>
<tr>
<td></td>
<td>Your process tried to cancel a package for which delivery was already in progress. The system will cancel further deliveries, but cannot guarantee that some recipient has not already received the package.</td>
</tr>
<tr>
<td>Error No./ (Field)</td>
<td>Text/ Meaning</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>4082 (RETN-CODE)</td>
<td>E-NO-RECIPS</td>
</tr>
<tr>
<td>4083 (RETN-CODE)</td>
<td>E-NOT-CREATED-BY-YOU</td>
</tr>
<tr>
<td>4084 (RETN-CODE)</td>
<td>E-PKG-NOT-SUBMITTED</td>
</tr>
<tr>
<td>4085 (RETN-CODE)</td>
<td>E-DATA-TOO-LONG</td>
</tr>
<tr>
<td>4086 (RETN-CODE)</td>
<td>W-DATASIZE-ADJUSTED</td>
</tr>
<tr>
<td>4087 (RETN-CODE)</td>
<td>E-INVALID-MAX-DATASIZE</td>
</tr>
</tbody>
</table>
### Table A-2. Error Codes (Continued)

<table>
<thead>
<tr>
<th>Error No./ (Field)</th>
<th>Text/ Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>4089 (RETN-CODE)</td>
<td>W-DATA-TRUNCATED</td>
</tr>
<tr>
<td>4092 (RETN-CODE)</td>
<td>E-INVALID-NUM-RQSTD</td>
</tr>
<tr>
<td>4093 (RETN-CODE)</td>
<td>E-SECURITY-VIOLATION</td>
</tr>
<tr>
<td>4094 (RETN-CODE)</td>
<td>E-PKG-CANCELED</td>
</tr>
<tr>
<td>4095 (RETN-CODE)</td>
<td>E-PKG-EXPIRED</td>
</tr>
<tr>
<td>4096 (RETN-CODE)</td>
<td>E-INVALID-COMPNT-TYPE</td>
</tr>
</tbody>
</table>
### Table A-2. Error Codes (Continued)

<table>
<thead>
<tr>
<th>Error No./ (Field)</th>
<th>Text/ Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>4100 (RETN-CODE)</td>
<td>W-ORD-KEY-TRUNCATED</td>
</tr>
<tr>
<td></td>
<td>An item was retrieved, but the corresponding ordering key was longer than the buffer space allocated for the ordering key in the response. The ordering key was truncated.</td>
</tr>
<tr>
<td>4101 (RETN-CODE)</td>
<td>W-MAX-KEY-LEN-ADJUSTED</td>
</tr>
<tr>
<td></td>
<td>Your process specified an odd number for MAX-KEY-LEN. You should specify an even number for this value so that all elements of the returned array are word aligned. When an odd number is specified, TRANSFER automatically increments the value by 1.</td>
</tr>
<tr>
<td>4102 (RETN-CODE)</td>
<td>E-INVALID-KEY-LEN</td>
</tr>
<tr>
<td></td>
<td>Your process specified an ordering key length that was less than 0 or greater than 200. The SCAN-FOLDER-BY-KEY UOW also returns this error if GENERIC-KEY-LEN is greater than STARTING-KEY-LEN.</td>
</tr>
<tr>
<td>4103 (RETN-CODE)</td>
<td>E-INCORRECT-FLD-ORDER</td>
</tr>
<tr>
<td></td>
<td>Your process issued a SAVE-ITEM UOW for a folder with an APPLIC-DEFINED ordering discipline; or your process issued either a SAVE-ITEM-BY-KEY or SCAN-FOLDER-BY-KEY UOW for a folder with an ordering discipline other than APPLIC-DEFINED.</td>
</tr>
<tr>
<td>4104 (RETN-CODE)</td>
<td>E-DUP-ORDER-KEY</td>
</tr>
<tr>
<td></td>
<td>The specified folder does not allow duplicate ordering keys. Your process attempted to save an item with an ordering key identical to the ordering key of another item already saved in the specified folder.</td>
</tr>
</tbody>
</table>
## Error Codes

**Table A-2. Error Codes (Continued)**

<table>
<thead>
<tr>
<th>Error No./ (Field)</th>
<th>Text/ Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>4105 (RETN-CODE)</td>
<td>E-CONCURRNT-FLD-UPDATE</td>
</tr>
<tr>
<td></td>
<td>Your process attempted to access a folder that was being updated at the same time. Either the folder was being deleted or the folder order was being altered. WARNING: If the UOW required a TMF transaction, your process must issue an ABORT-TRANSACTION; failure to issue an ABORT-TRANSACTION can cause the folder to become inconsistent.</td>
</tr>
<tr>
<td>4106 (RETN-CODE)</td>
<td>E-BAD-ORD-CRITERIA</td>
</tr>
<tr>
<td></td>
<td>The ordering criteria for the folder is invalid. Report this error to your Tandem representative. The folder is still usable and most UOWs accessing folders can still be issued for the folder; however, additional items cannot be stored in this folder.</td>
</tr>
<tr>
<td>4201 (RETN-CODE)</td>
<td>E-CONTEXT-ERR</td>
</tr>
<tr>
<td></td>
<td>A GUARDIAN error occurred while the current request was in process. This is an unusual event. In response, your process should abort the current transaction and retry the operation. Report this error to your Tandem representative.</td>
</tr>
<tr>
<td>4202 (RETN-CODE)</td>
<td>E-INVALID-APPL-CODE</td>
</tr>
<tr>
<td></td>
<td>A naming error occurred. This is an unusual condition. In response, your process should abort the current transaction and retry the operation.</td>
</tr>
</tbody>
</table>
### Table A-2. Error Codes (Continued)

<table>
<thead>
<tr>
<th>Error No./ (Field)</th>
<th>Text/ Meaning</th>
</tr>
</thead>
</table>
| 4205 (RETN-CODE) | E-MODEL-DEPOT-ABSENT  
Your process specified a model depot name that existed in the name directory, but for which no profile existed. |
| 4208 (RETN-CODE) | E-CORR-BLANK  
Your process attempted to add a new correspondent, but left the CORR-NAME field blank. |
| 4209 (IDENTIFIER-ERR) | E-INVALID-IDENTIFIER  
A request to obtain or update profile data elements included an identifier that TRANSFER could not recognize as legal. This error occurs only in response to the GET-PROFILE-ELEM and ALTER-PROFILE-ELEM UOWs, and is returned in the ELEM-RETN-CODE field rather than the RETN-CODE field. |
| 4210 (RETN-CODE) | W-IDENTIFIER-ERRS  
In a GET-PROFILE-ELEM or ALTER-PROFILE-ELEM UOW, your process included one or more identifiers that have an error associated with them. Examine the ELEM-RETN-CODE field(s), find the errors, correct them, and resubmit the UOW. |
| 4213 (RETN-CODE) | E-ITEMS-EXIST  
Your process issued a request to delete a folder or a depot, but directly or indirectly referenced a folder that contained one or more items. |
## Table A-2. Error Codes (Continued)

<table>
<thead>
<tr>
<th>Error No./ (Field)</th>
<th>Text/ Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>4214 (RETN-CODE)</td>
<td>E-INVALID-AGENT-NAME</td>
</tr>
<tr>
<td></td>
<td>In the ALTER-AGENT-SELECT UOW, your process specified an invalid agent name. For instance, the name might have contained embedded blanks or special characters not valid in a SCREEN COBOL program name.</td>
</tr>
<tr>
<td>4218 (RETN-CODE)</td>
<td>W-CONTENTS-PURGED</td>
</tr>
<tr>
<td></td>
<td>Your process attempted to delete (with the DELETE-FOLDER UOW) either the INBOX or WASTEBASKET folder. In this case, TRANSFER deletes the contents of the folder, but not the folder itself.</td>
</tr>
<tr>
<td>4219 (RETN-CODE)</td>
<td>E-SESSIONS-ACTIVE</td>
</tr>
<tr>
<td></td>
<td>Your process issued a DELETE-DEPOT UOW that requested deletion of a depot, but a session was active for the depot at the time of the request.</td>
</tr>
<tr>
<td>4220 (RETN-CODE)</td>
<td>W-EXACT-REC-READ</td>
</tr>
<tr>
<td></td>
<td>In response to a READ-PROFILE-REC UOW, the exact record requested by your process was read.</td>
</tr>
<tr>
<td>4221 (RETN-CODE)</td>
<td>W-NEXT-REC-READ</td>
</tr>
<tr>
<td></td>
<td>In response to a GET-AGENT-SELECT or READ-PROFILE-REC UOW, the next record requested by your process was read.</td>
</tr>
<tr>
<td>Error No./ (Field)</td>
<td>Text/ Meaning</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>4225 (RETN-CODE)</td>
<td>E-MEMBERS-EXIST</td>
</tr>
<tr>
<td></td>
<td>Your process issued a DELETE-DLIST UOW (or DELETE-DEPOT UOW requesting deletion of a distribution list), but the list contained members.</td>
</tr>
<tr>
<td>4227 (RETN-CODE)</td>
<td>W-ERR-ON-MEMBER</td>
</tr>
<tr>
<td></td>
<td>Your process issued an ADD-MEMBER UOW or a DELETE-MEMBER UOW that referenced one or more member names that were in error. For information about the exact error, see the detailed MEMBER-NAME fields.</td>
</tr>
<tr>
<td>4229 (RETN-CODE)</td>
<td>E-MUST-BE-EA</td>
</tr>
<tr>
<td></td>
<td>Your process tried to set to another value a flag that can only be set to E or A.</td>
</tr>
<tr>
<td>4230 (RETN-CODE)</td>
<td>E-MUST-BE-IWD</td>
</tr>
<tr>
<td></td>
<td>Your process tried to set to another value a flag that can only be set to I, W, or D.</td>
</tr>
<tr>
<td>4231 (RETN-CODE)</td>
<td>E-INVALID-AGENT-FLAG</td>
</tr>
<tr>
<td></td>
<td>Your process attempted to set an AGENT-FLAG to a value other than Y or N.</td>
</tr>
<tr>
<td>4232 (RETN-CODE)</td>
<td>E-INVALID-AGENT-TYPE</td>
</tr>
<tr>
<td></td>
<td>Your process attempted to set an AGENT-TYPE flag to a value other than Y or N.</td>
</tr>
</tbody>
</table>
## Table A-2. Error Codes (Continued)

<table>
<thead>
<tr>
<th>Error No./ (Field)</th>
<th>Text/ Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>4237 RETN-CODE</td>
<td>E-INVALID-GUARD-NAME</td>
</tr>
<tr>
<td></td>
<td>Your process attempted to reference a file with an invalid GUARDIAN file name.</td>
</tr>
<tr>
<td>4238 RETN-CODE</td>
<td>E-TFER-PROFILE-ABSENT</td>
</tr>
<tr>
<td></td>
<td>Your process referenced a correspondent for whom no TRANSFER profile record was present. If this occurs when your process is attempting to access the profile records of a logged-on correspondent, call your Tandem representative.</td>
</tr>
<tr>
<td>4239 RETN-CODE</td>
<td>E-MAIL-PROFILE-ABSENT</td>
</tr>
<tr>
<td></td>
<td>While using T/MAIL, a correspondent referenced a depot for which no T/MAIL profile record was present. If this occurs when your process is attempting to access the profile records of a logged-on correspondent, call your Tandem representative.</td>
</tr>
<tr>
<td>4240 RETN-CODE</td>
<td>E-FAX-PROFILE-ABSENT</td>
</tr>
<tr>
<td></td>
<td>Your process referenced a correspondent for whom no FAX profile record was present.</td>
</tr>
<tr>
<td>4241 RETN-CODE</td>
<td>E-TFER-CTL-REC-ABSENT</td>
</tr>
<tr>
<td></td>
<td>TRANSFER could not locate a required system control record. Call your Tandem representative.</td>
</tr>
<tr>
<td>4242 RETN-CODE</td>
<td>E-MAIL-CTL-REC-ABSENT</td>
</tr>
<tr>
<td></td>
<td>While interfacing with T/MAIL, TRANSFER could not locate a required system control record. Call your Tandem representative.</td>
</tr>
<tr>
<td>Error No./ (Field)</td>
<td>Text/ Meaning</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------</td>
</tr>
<tr>
<td>4265 RETN-CODE</td>
<td>E-INVALID-ORD-DISCIPLN</td>
</tr>
<tr>
<td></td>
<td>Your process entered a value other than T (TIME-SAVED), C (CREATOR-NAME), E (EARLIEST-DELIV-DATE), or A (APPLIC-DEFINED) for the folder ordering discipline.</td>
</tr>
<tr>
<td>4266 RETN-CODE</td>
<td>E-INVALID-APP-ORD-TYPE</td>
</tr>
<tr>
<td></td>
<td>Your process entered an invalid application defined ordering type. The value was outside the 0 through 9999 range.</td>
</tr>
<tr>
<td>4267 RETN-CODE</td>
<td>E-ITEMS-IN-FLD</td>
</tr>
<tr>
<td></td>
<td>Your process attempted to alter the ordering criteria of a folder that has items saved in it. The folder must be empty before the ordering criteria can be altered.</td>
</tr>
<tr>
<td>4268 RETN-CODE</td>
<td>E-SPECIAL-FLD</td>
</tr>
<tr>
<td></td>
<td>Your process attempted to alter the ordering criteria of the INBOX, OUTBOX, or WASTEBASKET to values invalid for a special folder. A special folder cannot have an APPLIC-DEFINED ordering discipline and must allow duplicate ordering keys.</td>
</tr>
<tr>
<td>4300 ELEM-RETNO-CODE</td>
<td>E-PASSWORD-MATCH-FAIL</td>
</tr>
<tr>
<td></td>
<td>In a request to alter a profile, your process used an invalid password that does not match that in the profile. This is a detailed identifier error, accompanied by UOW Warning No. 4210.</td>
</tr>
<tr>
<td>Error No./ (Field)</td>
<td>Text/ Meaning</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------</td>
</tr>
</tbody>
</table>
| 4302 (ELEM-RETN-CODE) | E-NOT-UPDATABLE  
Your process issued an ALTER-PROFILE-ELEM UOW with an invalid identifier. This is a detailed identifier error, accompanied by UOW Warning No. 4210. |
| 4303 (ELEM-RETN-CODE) | E-WINDOW-TOO-LARGE  
Your process specified a time window outside the allowed range for one of the following parameters: maximum package lifespan, package expiration, delivery end, minimum delivery window, or minimum expiration. When a package is submitted and the depot's default expiration or delivery values conflict with the system defaults, the values for the package are adjusted to be within the system limits. This is a detailed identifier error, accompanied by UOW Error No. 4210. |
| 4304 (ELEM-RETN-CODE) | E-INVALID-GUARDIAN-ID  
Your process specified a blank or invalid GUARDIAN-ID, or an invalid password. Enter a valid GUARDIAN-ID (which can be NONE) or password. This is a detailed identifier error, accompanied by UOW Error No. 4210. |
| 4305 (ELEM-RETN-CODE) | E-INVALID-FILENAME  
In an ALTER-PROFILE-ELEM UOW, your process specified an invalid GUARDIAN file name. Enter a name that is acceptable to GUARDIAN. This is a detailed identifier error, accompanied by UOW Error No. 4210. |
<table>
<thead>
<tr>
<th>Error No./ (Field)</th>
<th>Text/ Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>4306 (ELEM-RETN-CODE)</td>
<td>E-BAD-VOL-SUBVOL &lt;br&gt; Your process specified an invalid GUARDIAN file name. Enter a name that is acceptable to GUARDIAN. This is a detailed identifier error, accompanied by UOW Error No. 4210.</td>
</tr>
<tr>
<td>4307 (ELEM-RETN-CODE)</td>
<td>E-FIELD-TOO-LONG &lt;br&gt; Your process specified a field that is longer than the maximum field length allowed. Specify a shorter field. This is a detailed identifier error, accompanied by UOW Error No. 4210.</td>
</tr>
<tr>
<td>4308 (ELEM-RETN-CODE)</td>
<td>E-INVALID-NODE &lt;br&gt; Your process specified an invalid node name. Either the name format was incorrect or the node did not exist. This is a detailed identifier error, accompanied by UOW Error No. 4210.</td>
</tr>
<tr>
<td>4310 (ELEM-RETN-CODE)</td>
<td>E-PRIORITY-SEQUENCE &lt;br&gt; Your process specified priority numbers that were not in logical sequence (LOW greater than HIGH, for example). This is a detailed identifier error, accompanied by UOW Error No. 4210.</td>
</tr>
<tr>
<td>4311 (ELEM-RETN-CODE)</td>
<td>E-INVALID-EDITOR &lt;br&gt; Your process specified an invalid text editor file name.</td>
</tr>
</tbody>
</table>
Table A-2. Error Codes (Continued)

<table>
<thead>
<tr>
<th>Error No./ (Field)</th>
<th>Text/ Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>4381 (ELEM-RETN-CODE)</td>
<td>E-DISC-DOES-NOT-EXIST</td>
</tr>
<tr>
<td></td>
<td>Your process referenced the name of a disc that does not exist. This is a detailed identifier error, accompanied by UOW Error No. 4210.</td>
</tr>
<tr>
<td>4382 (ELEM-RETN-CODE)</td>
<td>E-MUST-BE-SLT</td>
</tr>
<tr>
<td></td>
<td>Your process tried to set the MESSAGE-SEQUENCE flag to a value other than S, L, or T. This is a detailed identifier error, accompanied by UOW Error No. 4210.</td>
</tr>
<tr>
<td>4383 (ELEM-RETN-CODE)</td>
<td>E-MUST-BE-TCN</td>
</tr>
<tr>
<td></td>
<td>Your process tried to set the CC-LIST flag to a value other than T, C, or N. This is a detailed identifier error, accompanied by UOW Error No. 4260.</td>
</tr>
<tr>
<td>4384 (ELEM-RETN-CODE)</td>
<td>E-MUST-BE-MD</td>
</tr>
<tr>
<td></td>
<td>Your process tried to set the date format flag to a value other than M or D. This is a detailed identifier error, accompanied by UOW Error No. 4210.</td>
</tr>
<tr>
<td>4388 (ELEM-RETN-CODE)</td>
<td>E-PRIV-MUST-BE-RWN</td>
</tr>
<tr>
<td></td>
<td>Your process tried to set the SYSTEM-ADMIN flag to a value other than R, W, or N. This is a detailed identifier error, accompanied by UOW Error No. 4210.</td>
</tr>
<tr>
<td>Error No./ (Field)</td>
<td>Text/ Meaning</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>4390 E-ELEM-RETN-CODE</td>
<td>E-FIILENAME-BLANK</td>
</tr>
<tr>
<td>4391 E-ELEM-RETN-CODE</td>
<td>E-MUST-BE-IS</td>
</tr>
<tr>
<td>4392 E-ELEM-RETN-CODE</td>
<td>E-MUST-BE-DISC</td>
</tr>
<tr>
<td>4395 E-ELEM-RETN-CODE</td>
<td>E-INVALID-INDENT</td>
</tr>
<tr>
<td>4600 DELIV-ERR in Recip Record</td>
<td>W-NOT-PROCESSED-HERE</td>
</tr>
</tbody>
</table>
### Error Codes

#### Table A-2. Error Codes (Continued)

<table>
<thead>
<tr>
<th>Error No./(Field)</th>
<th>Text/ Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>4601</td>
<td>W-XPORTED</td>
</tr>
<tr>
<td>(DELIV-ERR in Recip Record)</td>
<td>In the local list of remote recipients used by a TAREQ in transporting a package, recipients are grouped according to nodes. Whenever the TAREQ completes its scan of the recipients for a particular node, it transports the package to that node and flags all recipients in the local list with this message to indicate that the package has been sent to them.</td>
</tr>
<tr>
<td>4602</td>
<td>E-XPORT-FAILED</td>
</tr>
<tr>
<td>(DELIV-ERR in Recip Record)</td>
<td>The TAREQ could not transport the package for this recipient, probably because the package was nested too deeply.</td>
</tr>
<tr>
<td>4603</td>
<td>E-TOO-LATE-TO-XPORT</td>
</tr>
<tr>
<td>(DELIV-ERR in Recip Record)</td>
<td>The TAREQ could not transport the package to this recipient because the delivery window closed. In some cases, this situation could result in multiple instances of a TAREQ event package. For instance, if a TAREQ at Node A sends a package to Nodes B and C, and the TAREQs at those nodes detect that the package is to be transported to Node D, and the delivery window has closed, then TAREQs at Nodes B and C will both notify Node A with the same event package.</td>
</tr>
<tr>
<td>4604</td>
<td>W-FULLY-EXPANDED-DLIST</td>
</tr>
<tr>
<td>(DELIV-ERR in Recip Record)</td>
<td>After a TAREQ has expanded each distribution list in a recipient list, it flags that distribution list with this message.</td>
</tr>
</tbody>
</table>
### Error Codes (Continued)

<table>
<thead>
<tr>
<th>Error No./ (Field)</th>
<th>Text/ Meaning</th>
</tr>
</thead>
</table>
| 4605              | E-CANCELED-UNEXAMINED  
(DELIV-ERR in Recip Record) A TAREQ removed a package from the INBOX folder before the recipient examined the package. This message occurs when a process issues the CANCEL-PKG UOW. |
| 4607              | E-EXPIRED-UNEXAMINED  
(DELIV-ERR in Recip Record) The package was transported, but expired without being read by the recipient. |
| 4608              | E-TOO-LATE-TO-DELIV  
(DELIV-ERR in Recip Record) The TAREQ could not deliver the package to a recipient before the delivery window closed. |
| 4609              | E-INCONSISTENT-RECIP  
(DELIV-ERR in Recip Record) In the recipient record, inconsistent record types have been recorded. Call your Tandem Representative. |

**NOTE**

Errors 4610 through 4622 reflect errors passed between TRANSFER asynchronous processing components. These error codes are never returned to the user, but they do appear in the OUT and LOG files of the asynchronous processing components.

| 4610              | E-ASYNC-UNKNOWN-FUNC  
|                  | Unassigned error code. |
### Table A-2. Error Codes (Continued)

<table>
<thead>
<tr>
<th>Error No./ (Field)</th>
<th>Text/ Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>4611</td>
<td>E-ASYNC-UNKNOWN-FUNC&lt;br&gt;An interprocess message between two components contained an unknown function code. Call your Tandem representative.</td>
</tr>
<tr>
<td>4612</td>
<td>E-ASYNCH-MESSAGE-SHORT&lt;br&gt;An interprocess message between two components was too short. Call your Tandem representative.</td>
</tr>
<tr>
<td>4613</td>
<td>E-ASYNCH-MESSAGE-LONG&lt;br&gt;An interprocess message between two components was too long. Call your Tandem representative.</td>
</tr>
<tr>
<td>4614</td>
<td>E-ASYNCH-BAD-SEQUENCE&lt;br&gt;The component receiving an interprocess message determined from its function that the sender of the message was not in the state that the receiver anticipated. The components will resynchronize themselves.</td>
</tr>
<tr>
<td>4615</td>
<td>W-ASYNCH-PRE-TRECV&lt;br&gt;A remote TRANSFER system is running, but its TSCHED does not know how to handle a request to have a TRECV assigned. The remote function will be put back into the NET file and will be retried later. The remote system must be upgraded before the remote assignment can be carried out.</td>
</tr>
<tr>
<td>Error No./ (Field)</td>
<td>Text/ Meaning</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------</td>
</tr>
</tbody>
</table>
| 4616              | **W-ASYNCH-NO-IDLE-TRECV**  
A remote TRANSFER system is running, but all of its TREVCs are busy. The remote function will be put back into the NET file and will be retried later. |
| 4617              | **W-ASYNCH-CANT-REACH**  
A remote TRANSFER system is not running. The remote function will be put back into the NET file and will be retried later. |
| 4618              | **E-ASYNCH-SAY-AGAIN**  
The component receiving an interprocess message is momentarily incapable of handling it correctly; the sender of the interprocess message should immediately repeat it. |
| 4619              | **W-ASYNCH-NO-TRECV-WORK**  
At the direction of its TFRONT, a TRECV became available to its TSCHED for assignment; but the TSCHED had no assignment for the TRECV. The TRECV should continue waiting for word from its TFRONT. This can occur when two TSCHEDs have problems establishing a TWORK to TRECV connection. |
| 4620              | **E-ASYNCH-INVALID-SEQNO**  
The synchronization between two components has become confused. The components will correct the situation. |
### Error Codes

#### Table A-2. Error Codes (Continued)

<table>
<thead>
<tr>
<th>Error No./ (Field)</th>
<th>Text/ Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>4621</td>
<td>E-ASYNCH-BAD-PASSWORD</td>
</tr>
<tr>
<td></td>
<td>A component supplied the wrong password and cannot start its own session. The component will get a corrected password and try again to start the session. This can occur because some components use computed rather than stored passwords.</td>
</tr>
<tr>
<td>4622</td>
<td>E-ASYNCH-WRONG-REV</td>
</tr>
<tr>
<td></td>
<td>A component has determined that some other component is not at an acceptable revision level. Make sure that your configuration does not incorrectly mix components from various releases.</td>
</tr>
</tbody>
</table>

**NOTE**

Errors 4902 through 4924 reflect errors on files in the TRANSFER data base. The IPC-RETN-CODE-DETAIL field contains the specific file error code returned by the GUARDIAN operating system. File error codes are listed in Table A-1.

If your process receives one of these errors in response to a request issued within a TMF transaction, you MUST abort this transaction.

<table>
<thead>
<tr>
<th>4902</th>
<th>E-ERR-PROFILE-FILE</th>
</tr>
</thead>
<tbody>
<tr>
<td>(IPC-RETN-CODE or RETN-CODE)</td>
<td>An error occurred on the file named PROFILE, which contains profiles of the correspondents who have depots at the node.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4904</th>
<th>E-ERR-SESSION-FILE</th>
</tr>
</thead>
<tbody>
<tr>
<td>(IPC-RETN-CODE or RETN-CODE)</td>
<td>An error occurred on the file named SESSION, which defines all of the currently active sessions.</td>
</tr>
<tr>
<td>Error No./ (Field)</td>
<td>Text/ Meaning</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>4906</td>
<td>E-ERR-ITEMDESC-FILE</td>
</tr>
<tr>
<td>(IPC-RETN-CODE or RETN-CODE)</td>
<td>An error occurred on the file named ITEMDESC, which contains the item descriptors for all items and packages defined for the node, and for all packages received from other nodes.</td>
</tr>
<tr>
<td>4908</td>
<td>E-ERR-ITEMDATA-FILE</td>
</tr>
<tr>
<td>(IPC-RETN-CODE or RETN-CODE)</td>
<td>An error occurred on the file named ITEMDATA, which contains the data for all items and packages defined for the node, and for all packages received from other nodes.</td>
</tr>
<tr>
<td>4910</td>
<td>E-ERR-RECIP-FILE</td>
</tr>
<tr>
<td>(IPC-RETN-CODE or RETN-CODE)</td>
<td>An error occurred on the file named RECIP, which contains the names of recipients of packages either sent from or delivered to the node.</td>
</tr>
<tr>
<td>4912</td>
<td>E-ERR-FOLDER-FILE</td>
</tr>
<tr>
<td>(IPC-RETN-CODE or RETN-CODE)</td>
<td>An error occurred on the file named FOLDER, which contains the item IDs of items saved in each folder at the node.</td>
</tr>
<tr>
<td>4914</td>
<td>E-ERR-DLIST-FILE</td>
</tr>
<tr>
<td>(IPC-RETN-CODE or RETN-CODE)</td>
<td>An error occurred on the file named DLIST, which contains the names of members of distribution lists defined at the node.</td>
</tr>
<tr>
<td>Error No./ (Field)</td>
<td>Text/ Meaning</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>4916</td>
<td>E-ERR-READY-FILE</td>
</tr>
<tr>
<td>4918</td>
<td>E-ERR-TIME-FILE</td>
</tr>
<tr>
<td>4920</td>
<td>E-ERR-NET-FILE</td>
</tr>
<tr>
<td>4922</td>
<td>E-ERR-INV-FOLDER-FILE</td>
</tr>
<tr>
<td>4924</td>
<td>E-ERR-QUEUE-FILE</td>
</tr>
</tbody>
</table>
### Table A-2. Error Codes (Continued)

<table>
<thead>
<tr>
<th>Error No./ (Field)</th>
<th>Text/ Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>4990</td>
<td>E-IO-TIMEOUT</td>
</tr>
<tr>
<td>(IPC-RETN-CODE or RETN-CODE)</td>
<td>TISERV could not access portions of the TRANSFER data base needed to complete the requested operation because the data base was being modified by another correspondent. This error code can be returned by any UOW except NOOP. The RETN-CODE-DETAIL field contains the GUARDIAN file code for the file that was unavailable. Your application should abort or restart the current TMF transaction (if one was in progress when the error occurred); then, retry the request under a new transaction.</td>
</tr>
<tr>
<td>5600</td>
<td>E-CORR-NSRV-ERR</td>
</tr>
<tr>
<td>(RETN-CODE)</td>
<td>The name server encountered an error while processing a correspondent name. The RETN-CODE-DETAIL field contains the GUARDIAN file error code describing the problem.</td>
</tr>
<tr>
<td>5601</td>
<td>E-CORR-NOT-FOUND</td>
</tr>
<tr>
<td>(RETN-CODE)</td>
<td>The name server could not locate the correspondent referenced. This condition is analogous to placing a telephone call to a number that does not exist, or that is not listed in the telephone directory.</td>
</tr>
<tr>
<td>5602</td>
<td>E-CORR-BAD-NAME</td>
</tr>
<tr>
<td>(RETN-CODE)</td>
<td>A correspondent name contained a syntax error.</td>
</tr>
<tr>
<td>5603</td>
<td>E-CORR-BAD-TYPE</td>
</tr>
<tr>
<td>(RETN-CODE)</td>
<td>Your process referenced a correspondent name and the name was located by TRANSFER, but it was not of type CORRESPONDENT.</td>
</tr>
<tr>
<td>Error No./ (Field)</td>
<td>Text/ Meaning</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>5604 (RETN-CODE)</td>
<td>E-CORR-NO-SUCH-NODE</td>
</tr>
<tr>
<td></td>
<td>The node referenced by your process either does not exist, or was specified with an invalid name. This condition is analogous to placing a telephone call to an area code that does not exist.</td>
</tr>
<tr>
<td>5605 (RETN-CODE)</td>
<td>E-CORR-SECURITY</td>
</tr>
<tr>
<td></td>
<td>A security violation occurred in a reference to a correspondent.</td>
</tr>
<tr>
<td>5606 (RETN-CODE)</td>
<td>E-CORR-NSRV-NOT-FOUND</td>
</tr>
<tr>
<td></td>
<td>TRANSFER could not locate the correspondent's name server.</td>
</tr>
<tr>
<td>5607 (RETN-CODE)</td>
<td>E-CORR-NSRV-DOWN</td>
</tr>
<tr>
<td></td>
<td>The correspondent's name server is not operating correctly. The RETN-CODE-DETAIL field contains the GUARDIAN file error code describing the problem.</td>
</tr>
<tr>
<td>5608 (RETN-CODE)</td>
<td>E-CORR-NO-PARENT</td>
</tr>
<tr>
<td></td>
<td>When your process tried to add a correspondent, the name server could not locate the CORR directory.</td>
</tr>
<tr>
<td>5609 (RETN-CODE)</td>
<td>E-CORR-NOT-EMPTY</td>
</tr>
<tr>
<td></td>
<td>This error reflects difficulty with the name directory. Call your Tandem representative.</td>
</tr>
<tr>
<td>Error No./ (Field)</td>
<td>Text/ Meaning</td>
</tr>
<tr>
<td>-------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>5611 (RETN-CODE)</td>
<td>E-CORR-NET-DOWN</td>
</tr>
<tr>
<td></td>
<td>Processing of a correspondent or recipient name required access to names at other nodes, but the network was down. This condition is analogous to placing a telephone call to a number that is either disconnected or busy.</td>
</tr>
<tr>
<td>5612 (RETN-CODE)</td>
<td>E-CORR-ALREADY-EXISTS</td>
</tr>
<tr>
<td></td>
<td>With the CREATE-DEPOT or CREATE-DLIST UOW, your process tried to add a correspondent whose name already exists in the system.</td>
</tr>
<tr>
<td>5622 (RETN-CODE)</td>
<td>E-CORR-NOT-SAME-NODE</td>
</tr>
<tr>
<td></td>
<td>Your process requested an operation involving a correspondent at another node, but this operation is restricted to correspondents at the local node.</td>
</tr>
<tr>
<td>5623 (RETN-CODE)</td>
<td>E-CORR-AMBIGUOUS-NAME</td>
</tr>
<tr>
<td></td>
<td>Your process entered a name containing a wildcard character, but more characters are needed to make the name unique.</td>
</tr>
<tr>
<td>5624 (RETN-CODE)</td>
<td>E-CORR-BAD-SUFFIX</td>
</tr>
<tr>
<td></td>
<td>Your process specified a recipient name with an improper suffix. Perhaps the parentheses were not balanced, or the name contained illegal characters.</td>
</tr>
<tr>
<td>Error No./ (Field)</td>
<td>Text/ Meaning</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>5625 (RETN-CODE)</td>
<td>E-DLIST-NSRV-ERR</td>
</tr>
<tr>
<td></td>
<td>The name server encountered an error while processing a distribution list. The RETN-CODE-DETAIL field contains the GUARDIAN file error code describing the problem.</td>
</tr>
<tr>
<td>5626 (RETN-CODE)</td>
<td>E-DLIST-NOT-FOUND</td>
</tr>
<tr>
<td></td>
<td>The name server could not locate the distribution list referenced by your process. Either the list does not exist, or an improper name was supplied.</td>
</tr>
<tr>
<td>5627 (RETN-CODE)</td>
<td>E-DLIST-BAD-NAME</td>
</tr>
<tr>
<td></td>
<td>Your process entered either an invalid correspondent name, or a blank one. The name, for instance, might have contained embedded blanks or illegal characters.</td>
</tr>
<tr>
<td>5628 (RETN-CODE)</td>
<td>E-DLIST-BAD-TYPE</td>
</tr>
<tr>
<td></td>
<td>Your process entered a distribution list name and the name server located that name, but it represented an object of a type other than DISTRIBUTION-LIST.</td>
</tr>
<tr>
<td>5629 (RETN-CODE)</td>
<td>E-DLIST-NO-SUCH-NODE</td>
</tr>
<tr>
<td></td>
<td>Your process specified a distribution list with a node name that either was invalid or that referenced a nonexistent node.</td>
</tr>
<tr>
<td>5630 (RETN-CODE)</td>
<td>E-DLIST-SECURITY</td>
</tr>
<tr>
<td></td>
<td>A security violation occurred in a reference to a distribution list.</td>
</tr>
</tbody>
</table>
### Table A-2. Error Codes (Continued)

<table>
<thead>
<tr>
<th>Error No./ (Field)</th>
<th>Text/ Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>5631 (RETN-CODE)</td>
<td>E-DLIST-NSRV-NOT-FOUND</td>
</tr>
<tr>
<td></td>
<td>TRANSFER could not locate the distribution list's name server.</td>
</tr>
<tr>
<td>5632 (RETN-CODE)</td>
<td>E-DLIST-NSRV-DOWN</td>
</tr>
<tr>
<td></td>
<td>The distribution list's name server was not operating correctly. The RETN-CODE-DETAIL field contains the GUARDIAN file error code describing the problem.</td>
</tr>
<tr>
<td>5633 (RETN-CODE)</td>
<td>E-DLIST-NO-PARENT</td>
</tr>
<tr>
<td></td>
<td>Your process tried to add a distribution list under the name of a correspondent that could not be located. This error occurs, for instance, if the distribution list name contains periods.</td>
</tr>
<tr>
<td>5634 (RETN-CODE)</td>
<td>E-DLIST-NOT-EMPTY</td>
</tr>
<tr>
<td></td>
<td>Your process tried to delete a distribution list that contains named objects that have not yet been deleted.</td>
</tr>
<tr>
<td>5636 (RETN-CODE)</td>
<td>E-DLIST-NET-DOWN</td>
</tr>
<tr>
<td></td>
<td>Processing of a distribution list name required access to names at other nodes, but the network was down.</td>
</tr>
<tr>
<td>5637 (RETN-CODE)</td>
<td>E-DLIST-ALREADY-EXISTS</td>
</tr>
<tr>
<td></td>
<td>Your process tried to create a distribution list (with the CREATE-DLIST UOW) using a name that was already in use for a distribution list or folder in your depot. The distribution list was not created.</td>
</tr>
</tbody>
</table>
### Table A-2. Error Codes (Continued)

<table>
<thead>
<tr>
<th>Error No./ (Field)</th>
<th>Text/ Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>5647 (RETN-CODE)</td>
<td>E-DLIST-NOT-SAME-NODE</td>
</tr>
<tr>
<td></td>
<td>Your process requested an operation on a distribution list defined at another node, but this operation is restricted to distribution lists at your node.</td>
</tr>
<tr>
<td>5648 (RETN-CODE)</td>
<td>E-DLIST-AMBIGUOUS-NAME</td>
</tr>
<tr>
<td></td>
<td>Your process entered a distribution list name containing a wildcard character, but more characters were needed to make the name unique.</td>
</tr>
<tr>
<td>5649 (RETN-CODE)</td>
<td>E-DLIST-BAD-SUFFIX</td>
</tr>
<tr>
<td></td>
<td>Your process specified a distribution list name with a bad suffix. Perhaps the parentheses were not balanced or the name contained illegal characters.</td>
</tr>
<tr>
<td>5650 (MBR-RETN-CODE)</td>
<td>E-MBR-NSRV-ERR</td>
</tr>
<tr>
<td></td>
<td>The name server encountered an error while processing the ADD-MEMBER UOW to add a distribution list member.</td>
</tr>
<tr>
<td>5651 (MBR-RETN-CODE)</td>
<td>E-MBR-NOT-FOUND</td>
</tr>
<tr>
<td></td>
<td>Your process tried to add a member to a distribution list (with the ADD-MEMBER UOW) but the name server could not locate that correspondent. Either the correspondent did not exist in the system, or your process supplied an invalid correspondent name.</td>
</tr>
<tr>
<td>Error No./ (Field)</td>
<td>Text/ Meaning</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>5652 (MBR-RETN-CODE)</td>
<td>E-MBR-BAD-NAME In an attempt to add a member (ADD-MEMBER UOW) or delete a member (DELETE-MEMBER UOW) on a distribution list, your process specified either an invalid correspondent name or a blank one. The name might have contained embedded blanks or illegal characters.</td>
</tr>
<tr>
<td>5653 (MBR-RETN-CODE)</td>
<td>E-MBR-BAD-TYPE Your process specified the name of a distribution list member and the name server located that name, but it represented an object of a type not valid for use as a distribution list member.</td>
</tr>
<tr>
<td>5654 (MBR-RETN-CODE)</td>
<td>E-MBR-NO-SUCH-NODE In an attempt to add a member to a distribution list (ADD-MEMBER UOW), your process specified a distribution list member with a node name that either was invalid or that referenced a nonexistent node.</td>
</tr>
<tr>
<td>5655 (MBR-RETN-CODE)</td>
<td>E-MBR-SECURITY In an attempt to add a member to a distribution list (ADD-MEMBER UOW), a security violation occurred.</td>
</tr>
</tbody>
</table>
### Error Codes

**Table A-2. Error Codes (Continued)**

<table>
<thead>
<tr>
<th>Error No./(Field)</th>
<th>Text/ Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>5656</td>
<td><strong>E-MBR-NSRV-NOT-FOUND</strong></td>
</tr>
<tr>
<td>(MBR-RETN-CODE)</td>
<td>Your process attempted to add a member (ADD-MEMBER UOW) or delete a member (DELETE-MEMBER UOW) on a distribution list; the member was defined at another node and TRANSFER was not running at that node. Wait until the node becomes available, and then attempt to add or delete the name again.</td>
</tr>
<tr>
<td>5657</td>
<td><strong>E-MBR-NSRV-DOWN</strong></td>
</tr>
<tr>
<td>(MBR-RETN-CODE)</td>
<td>Your process tried to add a member to a distribution list (ADD-MEMBER UOW), but the distribution list member's name server was not operating correctly.</td>
</tr>
<tr>
<td>5658</td>
<td><strong>E-MBR-NO-PARENT</strong></td>
</tr>
<tr>
<td>(MBR-RETN-CODE)</td>
<td>Your process attempted to add a correspondent to a distribution list, but the correspondent could not be located. This error occurs, for instance, if the correspondent's name contains periods.</td>
</tr>
<tr>
<td>5659</td>
<td><strong>E-MBR-NOT-EMPTY</strong></td>
</tr>
<tr>
<td>(MBR-RETN-CODE)</td>
<td>This error reflects difficulty with the name directory. Call your Tandem representative.</td>
</tr>
<tr>
<td>5661</td>
<td><strong>E-MBR-NET-DOWN</strong></td>
</tr>
<tr>
<td>(MBR-RETN-CODE)</td>
<td>In an attempt to add a member to a distribution list (ADD-MEMBER UOW), processing of a distribution list member's name required access to names at other nodes, but the network was down.</td>
</tr>
<tr>
<td>Error No./ (Field)</td>
<td>Text/ Meaning</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>5662 (MBR-RETN-CODE)</td>
<td>E-MBR-ALREADY-EXISTS</td>
</tr>
<tr>
<td>5673 (MBR-RETN-CODE)</td>
<td>E-MBR-AMBIGUOUS-NAME</td>
</tr>
<tr>
<td>5674 (MBR-RETN-CODE)</td>
<td>E-MBR-BAD-SUFFIX</td>
</tr>
<tr>
<td>5675 (RETN-CODE)</td>
<td>E-FLD-NSRV-ERR</td>
</tr>
<tr>
<td>5676 (RETN-CODE)</td>
<td>E-FLD-NOT-FOUND</td>
</tr>
</tbody>
</table>

- **E-MBR-ALREADY-EXISTS**: Your process tried to add a correspondent to a distribution list (ADD-MEMBER UOW), but that correspondent was already defined as a member of the list.

- **E-MBR-AMBIGUOUS-NAME**: In a request to add or delete a member on a distribution list (ADD-MEMBER or DELETE-MEMBER UOW) your process entered the name of a member containing a wildcard character, but more characters are needed to make the name unique.

- **E-MBR-BAD-SUFFIX**: While adding a name to a distribution list (ADD-MEMBER UOW), your process specified the name of a correspondent with an improper suffix. Perhaps the suffix contained unbalanced parentheses or illegal characters.

- **E-FLD-NSRV-ERR**: The name server encountered an error while processing a folder name. The RETN-CODE-DETAIL field contains the GUARDIAN file error code describing the problem.

- **E-FLD-NOT-FOUND**: The name server could not locate the folder referenced by your process. Either the folder does not exist, or an improper name was supplied.
<table>
<thead>
<tr>
<th>Error No./ (Field)</th>
<th>Text/ Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>5677 (RETN-CODE)</td>
<td>E-FLD-BAD-NAME &lt;br&gt;Your process entered either an invalid folder name or a blank one. The name, for instance, might have contained embedded blanks or illegal characters.</td>
</tr>
<tr>
<td>5678 (RETN-CODE)</td>
<td>E-FLD-BAD-TYPE &lt;br&gt;Your process entered a folder name and the name server located that name, but it represented an object other than a folder.</td>
</tr>
<tr>
<td>5679 (RETN-CODE)</td>
<td>E-FLD-NO-SUCH-NODE &lt;br&gt;Your process specified a folder with a node name that either was invalid or that referenced a nonexistent node.</td>
</tr>
<tr>
<td>5680 (RETN-CODE)</td>
<td>E-FLD-SECURITY &lt;br&gt;A security violation occurred in a reference to a folder.</td>
</tr>
<tr>
<td>5681 (RETN-CODE)</td>
<td>E-FLD-NSRV-NOT-FOUND &lt;br&gt;TRANSFER could not locate the folder's name server.</td>
</tr>
<tr>
<td>5682 (RETN-CODE)</td>
<td>E-FLD-NSRV-DOWN &lt;br&gt;The folder's name server was not operating correctly. The RETN-CODE-DETAIL field contains the GUARDIAN file error code describing the problem.</td>
</tr>
<tr>
<td>Error No./ (Field)</td>
<td>Text/ Meaning</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------</td>
</tr>
</tbody>
</table>
| 5683 (RETN-CODE) | **E-FLD-NO-PARENT**  
Your process tried to add a folder to a depot (correspondent name) that could not be located. This error occurs, for instance, if the folder name contains periods. |
| 5684 (RETN-CODE) | **E-FLD-NOT-EMPTY**  
This error indicates difficulty with the name directory. Call your Tandem representative. |
| 5686 (RETN-CODE) | **E-FLD-NET-DOWN**  
Processing of a folder name required access to names at other nodes, but the network was down. |
| 5687 (RETN-CODE) | **E-FLD-ALREADY-EXISTS**  
Your process tried to create a folder and used a name that was already in use for a folder or distribution list in your depot. The folder was not created. |
| 5697 (RETN-CODE) | **E-FLD-NOT-SAME-NODE**  
Your process requested an operation involving a folder at another node, but this operation is restricted to folders at the local node. |
| 5698 (RETN-CODE) | **E-FLD-AMBIGUOUS-NAME**  
Your process entered a name containing a wildcard character, but more characters are needed to make the name unique. |
Error Codes

Table A-2. Error Codes (Continued)

<table>
<thead>
<tr>
<th>Error No./ (Field)</th>
<th>Text/ Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>5700 (RETN-CODE)</td>
<td>E-ALIAS-NSRV-ERR</td>
</tr>
<tr>
<td></td>
<td>The name server encountered an error while processing a name input in the GET-CONFIG-NAME UOW. The RETN-CODE-DETAIL field contains the GUARDIAN file error code describing the problem.</td>
</tr>
<tr>
<td>5701 (RETN-CODE)</td>
<td>E-ALIAS-NOT-FOUND</td>
</tr>
<tr>
<td></td>
<td>The name server could not locate a name input with the GET-CONFIG-NAME UOW.</td>
</tr>
<tr>
<td>5702 (RETN-CODE)</td>
<td>E-ALIAS-BAD-NAME</td>
</tr>
<tr>
<td></td>
<td>The name server could not locate the name specified in the GET-CONFIG-NAME UOW.</td>
</tr>
<tr>
<td>5703 (RETN-CODE)</td>
<td>E-ALIAS-BAD-TYPE</td>
</tr>
<tr>
<td></td>
<td>Your process referenced a name in the GET-CONFIG-NAME UOW and the name server located that name, but it represented an object of a type not valid for the context in which it was referenced.</td>
</tr>
<tr>
<td>5704 (RETN-CODE)</td>
<td>E-ALIAS-NO-SUCH-NODE</td>
</tr>
<tr>
<td></td>
<td>Your process specified, in the GET-CONFIG-NAME UOW, an object with a node name that either was invalid or that referenced a nonexistent node.</td>
</tr>
<tr>
<td>5705 (RETN-CODE)</td>
<td>E-ALIAS-SECURITY</td>
</tr>
<tr>
<td></td>
<td>A security violation occurred in a reference to a name input in the GET-CONFIG-NAME UOW.</td>
</tr>
</tbody>
</table>
### Table A-2. Error Codes (Continued)

<table>
<thead>
<tr>
<th>Error No./ (Field)</th>
<th>Text/ Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>5706</td>
<td>E-ALIAS-NSRV-NOT-FOUND</td>
</tr>
<tr>
<td>(RETN-CODE)</td>
<td>TRANSFER could not locate the name server for an object referenced in a GET-CONFIG-NAME UOW.</td>
</tr>
<tr>
<td>5707</td>
<td>E-ALIAS-NSRV-DOWN</td>
</tr>
<tr>
<td>(RETN-CODE)</td>
<td>The name server for an object referenced by the GET-CONFIG-NAME UOW was not operating correctly. The RETN-CODE-DETAIL field contains the GUARDIAN file error describing the problem.</td>
</tr>
<tr>
<td>5708</td>
<td>E-ALIAS-NO-PARENT</td>
</tr>
<tr>
<td>(RETN-CODE)</td>
<td>A system error occurred. Call your Tandem representative.</td>
</tr>
<tr>
<td>5709</td>
<td>E-ALIAS-NOT-EMPTY</td>
</tr>
<tr>
<td>(RETN-CODE)</td>
<td>A system error occurred. Call your Tandem Representative.</td>
</tr>
<tr>
<td>5710</td>
<td>E-ALIAS-ALIAS</td>
</tr>
<tr>
<td>(RETN-CODE)</td>
<td>An ambiguous reference occurred in a name input with the GET-CONFIG-NAME UOW.</td>
</tr>
<tr>
<td>5711</td>
<td>E-ALIAS-NET-DOWN</td>
</tr>
<tr>
<td>(RETN-CODE)</td>
<td>Processing of a name entered with the GET-CONFIG-NAME UOW required access to names defined at other nodes, but the network was down.</td>
</tr>
</tbody>
</table>
### Table A-2. Error Codes (Continued)

<table>
<thead>
<tr>
<th>Error No./ (Field)</th>
<th>Text/ Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>5722</td>
<td>E-ALIAS-NOT-SAME-NODE</td>
</tr>
<tr>
<td>(RETN-CODE)</td>
<td>The GET-CONFIG-NAME UOW requested an operation involving an object defined at another node, but this operation is restricted to objects at this node.</td>
</tr>
<tr>
<td>5723</td>
<td>E-ALIAS-AMBIGUOUS-NAME</td>
</tr>
<tr>
<td>(RETN-CODE)</td>
<td>The GET-CONFIG-NAME UOW referenced a name containing a wildcard character, but more characters are needed to make the name unique.</td>
</tr>
<tr>
<td>5725</td>
<td>E-MODEL-NSRV-ERR</td>
</tr>
<tr>
<td>(RETN-CODE)</td>
<td>The name server encountered an error while processing a model name. The RETN-CODE-DETAIL field contains the GUARDIAN file error code describing the problem.</td>
</tr>
<tr>
<td>5726</td>
<td>E-MODEL-NOT-FOUND</td>
</tr>
<tr>
<td>(RETN-CODE)</td>
<td>Your process issued a CREATE-DEPOT-UOW, but the name server could not locate the model depot referenced by your process. Either the depot does not exist, or an improper name was supplied.</td>
</tr>
<tr>
<td>5727</td>
<td>E-MODEL-BAD-NAME</td>
</tr>
<tr>
<td>(RETN-CODE)</td>
<td>Your process issued a CREATE-DEPOT UOW and referenced either an invalid model depot name or a blank one. The name, for instance, might have contained embedded blanks or illegal characters.</td>
</tr>
</tbody>
</table>
Table A-2. Error Codes (Continued)

<table>
<thead>
<tr>
<th>Error No./ (Field)</th>
<th>Text/ Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>5728</td>
<td>E-MODEL-BAD-TYPE</td>
</tr>
<tr>
<td>(RETN-CODE)</td>
<td>Your process referenced a model depot name and the name server located that name, but it represented an object of a type not valid for use as a model depot.</td>
</tr>
<tr>
<td>5729</td>
<td>E-MODEL-NO-SUCH-NODE</td>
</tr>
<tr>
<td>(RETN-CODE)</td>
<td>In a CREATE-DEPOT UOW, your process specified a model depot with a node name that either was invalid or that referenced a nonexistent node.</td>
</tr>
<tr>
<td>5730</td>
<td>E-MODEL-SECURITY</td>
</tr>
<tr>
<td>(RETN-CODE)</td>
<td>A security violation occurred in a reference to a model depot.</td>
</tr>
<tr>
<td>5731</td>
<td>E-MODEL-NSRV-NOT-FOUND</td>
</tr>
<tr>
<td>(RETN-CODE)</td>
<td>Your process entered a CREATE-DEPOT UOW, but TRANSFER could not locate the model depot's name server.</td>
</tr>
<tr>
<td>5732</td>
<td>E-MODEL-NSRV-DOWN</td>
</tr>
<tr>
<td>(RETN-CODE)</td>
<td>Your process entered a CREATE-DEPOT UOW, but the model depot's name server was not operating correctly. The RETN-CODE-DETAIL field contains the GUARDIAN file error code describing the problem.</td>
</tr>
<tr>
<td>5733</td>
<td>E-MODEL-NO-PARENT</td>
</tr>
<tr>
<td>(RETN-CODE)</td>
<td>A system error occurred. Call your Tandem representative.</td>
</tr>
</tbody>
</table>
## Error Codes

### Table A-2. Error Codes (Continued)

<table>
<thead>
<tr>
<th>Error No./ (Field)</th>
<th>Text/ Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>5734 (RETN-CODE)</td>
<td>E-MODEL-NOT-EMPTY</td>
</tr>
<tr>
<td></td>
<td>A system error occurred. Call your Tandem representative.</td>
</tr>
<tr>
<td>5736 (RETN-CODE)</td>
<td>E-MODEL-NET-DOWN</td>
</tr>
<tr>
<td></td>
<td>Your process entered a CREATE-DEPOT UOW and processing of a model depot name required access to names at other nodes, but the network was down.</td>
</tr>
<tr>
<td>5737 (RETN-CODE)</td>
<td>E-MODEL-ALREADY-EXISTS</td>
</tr>
<tr>
<td></td>
<td>A system error occurred. Call your Tandem representative.</td>
</tr>
<tr>
<td>5747 (RETN-CODE)</td>
<td>E-MODEL-NOT-SAME-NODE</td>
</tr>
<tr>
<td></td>
<td>In a CREATE-DEPOT UOW, your process referenced a model depot located at another node.</td>
</tr>
<tr>
<td>5748 (RETN-CODE)</td>
<td>E-MODEL-AMBIGUOUS-NAME</td>
</tr>
<tr>
<td></td>
<td>In a CREATE-DEPOT UOW, your process referenced a model depot name containing a wildcard character, but more characters are needed to make the name unique.</td>
</tr>
<tr>
<td>5750 (RETN-CODE)</td>
<td>E-RECIP-NSRV-ERR</td>
</tr>
<tr>
<td></td>
<td>The name server encountered an error while processing a recipient list name. The RETN-CODE-DETAIL field contains the GUARDIAN file error code describing the problem.</td>
</tr>
<tr>
<td>Error No./ (Field)</td>
<td>Text/ Meaning</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>5751 (RETN-CODE)</td>
<td>E-RECIP-NOT-FOUND</td>
</tr>
<tr>
<td>5752 (RETN-CODE)</td>
<td>E-RECIP-BAD-NAME</td>
</tr>
<tr>
<td>5753 (RETN-CODE)</td>
<td>E-RECIP-BAD-TYPE</td>
</tr>
<tr>
<td>5754 (RETN-CODE)</td>
<td>E-RECIP-NO-SUCH-NODE</td>
</tr>
<tr>
<td>5755 (RETN-CODE)</td>
<td>E-RECIP-SECURITY</td>
</tr>
<tr>
<td>5756 (RETN-CODE)</td>
<td>E-RECIP-NSRV-NOT-FOUND</td>
</tr>
</tbody>
</table>
### Table A-2. Error Codes (Continued)

<table>
<thead>
<tr>
<th>Error No./ (Field)</th>
<th>Text/ Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>5757 (RETN-CODE)</td>
<td>E-RECIP-NSRV-DOWN &lt;br&gt;The recipient's name server was not operating correctly. The RETN-CODE-DETAIL field contains the GUARDIAN file error code describing the problem.</td>
</tr>
<tr>
<td>5758 (RETN-CODE)</td>
<td>E-RECIP-NO-PARENT &lt;br&gt;A system error occurred. Call your Tandem representative.</td>
</tr>
<tr>
<td>5759 (RETN-CODE)</td>
<td>E-RECIP-NOT-EMPTY &lt;br&gt;A system error occurred. Call your Tandem representative.</td>
</tr>
<tr>
<td>5761 (RETN-CODE)</td>
<td>E-RECIP-NET-DOWN &lt;br&gt;Processing of a recipient name required access to names at other nodes, but the network was down.</td>
</tr>
<tr>
<td>5762 (RETN-CODE)</td>
<td>E-RECIP-ALREADY-EXISTS &lt;br&gt;A system error occurred. Call your Tandem representative.</td>
</tr>
<tr>
<td>5773 (RETN-CODE)</td>
<td>E-RECIP-AMBIGUOUS-NAME &lt;br&gt;Your process entered a recipient name containing a wildcard character, but more characters are needed to make the name unique.</td>
</tr>
<tr>
<td>Error No./Field</td>
<td>Text/ Meaning</td>
</tr>
<tr>
<td>----------------</td>
<td>----------------</td>
</tr>
<tr>
<td>5774 RETN-CODE</td>
<td>E-RECIP-BAD-SUFFIX</td>
</tr>
<tr>
<td></td>
<td>Your process specified a recipient name with an improper suffix. Perhaps the parentheses were not balanced, or the name contained illegal characters.</td>
</tr>
<tr>
<td>6001 RETN-CODE</td>
<td>W-QUEUE-EMPTY</td>
</tr>
<tr>
<td></td>
<td>Returned by DEQ. There are no entries on the Queue file for the specified QUEUE-NAME.</td>
</tr>
<tr>
<td>6002 RETN-CODE</td>
<td>E-TOO-MANY-WAITERS</td>
</tr>
<tr>
<td></td>
<td>Returned by WAITQ. The Wait Manager cannot handle additional waiting requesters. The Wait Manager should be configured with a larger value for the MAXWAITERS parameters.</td>
</tr>
<tr>
<td>6003 RETN-CODE</td>
<td>E-INVALID-WAITPRIORITY</td>
</tr>
<tr>
<td></td>
<td>Returned by WAITQ. The value for WAIT-PRIORITY is not 0 through 199.</td>
</tr>
<tr>
<td>6004 RETN-CODE</td>
<td>E-INVALID-WAITTIMEOUT</td>
</tr>
<tr>
<td></td>
<td>Returned by WAITQ. The value for WAIT-TIMEOUT is not -1 through +9999.</td>
</tr>
<tr>
<td>6005 RETN-CODE</td>
<td>W-WAIT-TIMEOUT</td>
</tr>
<tr>
<td></td>
<td>Returned by WAITQ. The WAITQ UOW has been responded to because of elapsed time, not because of a new entry on the Queue file.</td>
</tr>
<tr>
<td>Error No./ (Field)</td>
<td>Text/ Meaning</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------------</td>
</tr>
</tbody>
</table>
| 6006 (IPC-RETN-CODE or RETN-CODE) | E-WAITMANAGER-UNAVAIL

Returned by ENQ. There is an error in communicating with the Wait Manager. Applications can use the NOTIFY-WAIT-MANAGER flag to add the queue entry if the application is willing to make the entry when the Wait Manager is unavailable.
### Error Codes

<table>
<thead>
<tr>
<th>Text</th>
<th>Error No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALIAS-ALIAS</td>
<td>5710</td>
</tr>
<tr>
<td>ALIAS-AMBIGUOUS-NAME</td>
<td>5723</td>
</tr>
<tr>
<td>ALIAS-BAD-NAME</td>
<td>5702</td>
</tr>
<tr>
<td>ALIAS-BAD-TYPE</td>
<td>5703</td>
</tr>
<tr>
<td>ALIAS-NET-DOWN</td>
<td>5711</td>
</tr>
<tr>
<td>ALIAS-NO-PARENT</td>
<td>5708</td>
</tr>
<tr>
<td>ALIAS-NO-SUCH-NODE</td>
<td>5704</td>
</tr>
<tr>
<td>ALIAS-NOT-EMPTY</td>
<td>5709</td>
</tr>
<tr>
<td>ALIAS-NOT-FOUND</td>
<td>5701</td>
</tr>
<tr>
<td>ALIAS-NOT-SAME-NODE</td>
<td>5722</td>
</tr>
<tr>
<td>ALIAS-NSRV-DOWN</td>
<td>5707</td>
</tr>
<tr>
<td>ALIAS-NSRV-ERR</td>
<td>5700</td>
</tr>
<tr>
<td>ALIAS-NSRV-NOT-FOUND</td>
<td>5706</td>
</tr>
<tr>
<td>ALIAS-SECURITY</td>
<td>5705</td>
</tr>
<tr>
<td>ALREADY-IN-SESSION</td>
<td>4019</td>
</tr>
<tr>
<td>ASYNCH-BAD-PASSWORD</td>
<td>4621</td>
</tr>
<tr>
<td>ASYNCH-BAD-SEQUENCE</td>
<td>4614</td>
</tr>
<tr>
<td>ASYNCH-CANT-REACH</td>
<td>4617</td>
</tr>
<tr>
<td>ASYNCH-INVALID-SEQNO</td>
<td>4620</td>
</tr>
<tr>
<td>ASYNCH-MESSAGE-LONG</td>
<td>4613</td>
</tr>
<tr>
<td>ASYNCH-MESSAGE-SHORT</td>
<td>4612</td>
</tr>
<tr>
<td>ASYNCH-NO-IDLE-TRECV</td>
<td>4616</td>
</tr>
<tr>
<td>ASYNCH-NO-TRECV-WORK</td>
<td>4619</td>
</tr>
<tr>
<td>ASYNCH-PRE-TRECV</td>
<td>4615</td>
</tr>
<tr>
<td>ASYNCH-SAY-AGAIN</td>
<td>4618</td>
</tr>
<tr>
<td>ASYNC-UNKNOWN-FUNC</td>
<td>4610</td>
</tr>
<tr>
<td>ASYNC-UNKNOWN-FUNC</td>
<td>4611</td>
</tr>
<tr>
<td>ASYNCH-WRONG-REV</td>
<td>4622</td>
</tr>
<tr>
<td>BAD-ITEM-DESCR</td>
<td>4040</td>
</tr>
<tr>
<td>BAD-ORD-CRITERIA</td>
<td>4106</td>
</tr>
<tr>
<td>BAD-TRANSACTION</td>
<td>4010</td>
</tr>
<tr>
<td>BAD-VOL-SUBVOL</td>
<td>4306</td>
</tr>
<tr>
<td>CANCELED-UNEXAMINED</td>
<td>4605</td>
</tr>
<tr>
<td>COMPNT-CYCLE</td>
<td>4038</td>
</tr>
<tr>
<td>COMPNT-NOT-FOUND</td>
<td>4039</td>
</tr>
<tr>
<td>CONCURRENT-SESSION</td>
<td>4005</td>
</tr>
<tr>
<td>CONCURRENT-FLD-UPDATE</td>
<td>4105</td>
</tr>
<tr>
<td>CONTENTS-PURGED</td>
<td>4218</td>
</tr>
<tr>
<td>CONTEXT-ERR</td>
<td>4201</td>
</tr>
<tr>
<td>CORR-ALREADY-EXISTS</td>
<td>5612</td>
</tr>
<tr>
<td>CORR-AMBIGUOUS-NAME</td>
<td>5623</td>
</tr>
<tr>
<td>CORR-BAD-NAME</td>
<td>5602</td>
</tr>
<tr>
<td>CORR-BAD-SUFFIX</td>
<td>5624</td>
</tr>
</tbody>
</table>
### Error Codes

**Table A-3. Alphabetic Listing of Error Codes (Continued)**

<table>
<thead>
<tr>
<th>Text</th>
<th>Error No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORR-BAD-TYPE</td>
<td>5603</td>
</tr>
<tr>
<td>CORR-BLANK</td>
<td>4208</td>
</tr>
<tr>
<td>CORR-NET-DOWN</td>
<td>5611</td>
</tr>
<tr>
<td>CORR-NO-PARENT</td>
<td>5608</td>
</tr>
<tr>
<td>CORR-NO-SUCH-NODE</td>
<td>5604</td>
</tr>
<tr>
<td>CORR-NOT-EMPTY</td>
<td>5609</td>
</tr>
<tr>
<td>CORR-NOT-FOUND</td>
<td>5601</td>
</tr>
<tr>
<td>CORR-NOT-SAME-NODE</td>
<td>5622</td>
</tr>
<tr>
<td>CORR-NSRV-DOWN</td>
<td>5607</td>
</tr>
<tr>
<td>CORR-NSRV-ERR</td>
<td>5600</td>
</tr>
<tr>
<td>CORR-NSRV-NOT-FOUND</td>
<td>5606</td>
</tr>
<tr>
<td>CORR-SECURITY</td>
<td>5605</td>
</tr>
<tr>
<td>DATA-TOO-LONG</td>
<td>4085</td>
</tr>
<tr>
<td>DATA-TRUNCATED</td>
<td>4089</td>
</tr>
<tr>
<td>DATASIZE-ADJUSTED</td>
<td>4086</td>
</tr>
<tr>
<td>DELIV-IN-PROGRESS</td>
<td>4081</td>
</tr>
<tr>
<td>DISC-DOES-NOT-EXIST</td>
<td>4381</td>
</tr>
<tr>
<td>DLIST-ALREADY-EXISTS</td>
<td>5637</td>
</tr>
<tr>
<td>DLIST-AMBIGUOUS-NAME</td>
<td>5648</td>
</tr>
<tr>
<td>DLIST-BAD-NAME</td>
<td>5627</td>
</tr>
<tr>
<td>DLIST-BAD-SUFFIX</td>
<td>5649</td>
</tr>
<tr>
<td>DLIST-BAD-TYPE</td>
<td>5628</td>
</tr>
<tr>
<td>DLIST-NET-DOWN</td>
<td>5636</td>
</tr>
<tr>
<td>DLIST-NO-PARENT</td>
<td>5633</td>
</tr>
<tr>
<td>DLIST-NO-SUCH-NODE</td>
<td>5629</td>
</tr>
<tr>
<td>DLIST-NO-EMPTY</td>
<td>5634</td>
</tr>
<tr>
<td>DLIST-NO-FOUND</td>
<td>5626</td>
</tr>
<tr>
<td>DLIST-NO-SAME-NODE</td>
<td>5647</td>
</tr>
<tr>
<td>DLIST-NSRV-DOWN</td>
<td>5632</td>
</tr>
<tr>
<td>DLIST-NSRV-ERR</td>
<td>5625</td>
</tr>
<tr>
<td>DLIST-NSRV-NOT-FOUND</td>
<td>5631</td>
</tr>
<tr>
<td>DLIST-SECURITY</td>
<td>5630</td>
</tr>
<tr>
<td>DUP-ORDER-KEY</td>
<td>4104</td>
</tr>
<tr>
<td>EOF</td>
<td>4001</td>
</tr>
<tr>
<td>ERR-DLIST-FILE</td>
<td>4914</td>
</tr>
<tr>
<td>ERR-FOLDER-FILE</td>
<td>4912</td>
</tr>
<tr>
<td>ERR-INV-FOLDER-FILE</td>
<td>4922</td>
</tr>
<tr>
<td>ERR-ITEMDATA-FILE</td>
<td>4908</td>
</tr>
<tr>
<td>ERR-ITEMDESC-FILE</td>
<td>4906</td>
</tr>
<tr>
<td>ERR-NET-FILE</td>
<td>4920</td>
</tr>
<tr>
<td>ERR-ON-MEMBER</td>
<td>4227</td>
</tr>
<tr>
<td>ERR-PROFILE-FILE</td>
<td>4902</td>
</tr>
<tr>
<td>ERR-QUEUE-FILE</td>
<td>4924</td>
</tr>
</tbody>
</table>
### Table A-3. Alphabetic Listing of Error Codes (Continued)

<table>
<thead>
<tr>
<th>Text</th>
<th>Error No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERR-READY-FILE</td>
<td>4916</td>
</tr>
<tr>
<td>ERR-RECIP-FILE</td>
<td>4910</td>
</tr>
<tr>
<td>ERR-SESSION-FILE</td>
<td>4904</td>
</tr>
<tr>
<td>ERR-TIME-FILE</td>
<td>4918</td>
</tr>
<tr>
<td>EXACT-REC-READ</td>
<td>4220</td>
</tr>
<tr>
<td>EXPIRED-UNEXAMINED</td>
<td>4607</td>
</tr>
<tr>
<td>FAX-PROF-ABSENT</td>
<td>4240</td>
</tr>
<tr>
<td>FIELD-TOO-LONG</td>
<td>4307</td>
</tr>
<tr>
<td>FILENAME-BLANK</td>
<td>4390</td>
</tr>
<tr>
<td>FLD-ALREADY-EXISTS</td>
<td>5687</td>
</tr>
<tr>
<td>FLD-AMBIGUOUS-NAME</td>
<td>5698</td>
</tr>
<tr>
<td>FLD-BAD-NAME</td>
<td>5677</td>
</tr>
<tr>
<td>FLD-BAD-TYPE</td>
<td>5678</td>
</tr>
<tr>
<td>FLD-NET-DOWN</td>
<td>5686</td>
</tr>
<tr>
<td>FLD-NO-PARENT</td>
<td>5683</td>
</tr>
<tr>
<td>FLD-NO-SUCH-NODE</td>
<td>5679</td>
</tr>
<tr>
<td>FLD-NOT-EMPTY</td>
<td>5684</td>
</tr>
<tr>
<td>FLD-NOT-FOUND</td>
<td>5676</td>
</tr>
<tr>
<td>FLD-NOT-SAME-NODE</td>
<td>5697</td>
</tr>
<tr>
<td>FLD-NSRV-DOWN</td>
<td>5682</td>
</tr>
<tr>
<td>FLD-NSRV-ERR</td>
<td>5675</td>
</tr>
<tr>
<td>FLD-NSRV-NOT-FOUND</td>
<td>5681</td>
</tr>
<tr>
<td>FLD-SECURITY</td>
<td>5680</td>
</tr>
<tr>
<td>FULLY-EXPANDED-DLIST</td>
<td>4604</td>
</tr>
<tr>
<td>IDENTIFIER-ERRS</td>
<td>4210</td>
</tr>
<tr>
<td>INCONSISTENT-RECIP</td>
<td>4609</td>
</tr>
<tr>
<td>INCORRECT-FLD-ORDER</td>
<td>4103</td>
</tr>
<tr>
<td>INVALID-AGENT-FLAG</td>
<td>4231</td>
</tr>
<tr>
<td>INVALID-AGENT-NAME</td>
<td>4214</td>
</tr>
<tr>
<td>INVALID-AGENT-SEL</td>
<td>4054</td>
</tr>
<tr>
<td>INVALID-AGENT-TYPE</td>
<td>4232</td>
</tr>
<tr>
<td>INVALID-APP-ORD-TYPE</td>
<td>4266</td>
</tr>
<tr>
<td>INVALID-APPL-CODE</td>
<td>4202</td>
</tr>
<tr>
<td>INVALID-APPLIC-ID</td>
<td>4055</td>
</tr>
<tr>
<td>INVALID-COMPNT-TYPE</td>
<td>4096</td>
</tr>
<tr>
<td>INVALID-DATE-TIME</td>
<td>4073</td>
</tr>
<tr>
<td>INVALID-EDITOR</td>
<td>4311</td>
</tr>
<tr>
<td>INVALID-FIILENAME</td>
<td>4305</td>
</tr>
<tr>
<td>INVALID-GUARD-NAME</td>
<td>4237</td>
</tr>
<tr>
<td>INVALID-GUARDIAN-ID</td>
<td>4304</td>
</tr>
<tr>
<td>INVALID-IDENTIFIER</td>
<td>4209</td>
</tr>
<tr>
<td>INVALID-INDENT</td>
<td>4395</td>
</tr>
<tr>
<td>INVALID-ITEM-TYPE</td>
<td>4056</td>
</tr>
</tbody>
</table>
Table A-3. Alphabetic Listing of Error Codes (Continued)

<table>
<thead>
<tr>
<th>Text</th>
<th>Error No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVALID-KEY-LEN</td>
<td>4102</td>
</tr>
<tr>
<td>INVALID-MAX-DATASIZE</td>
<td>4087</td>
</tr>
<tr>
<td>INVALID-NODE</td>
<td>4398</td>
</tr>
<tr>
<td>INVALID-NUM-RQSTD</td>
<td>4092</td>
</tr>
<tr>
<td>INVALID-ORD-DISCIPLN</td>
<td>4265</td>
</tr>
<tr>
<td>INVALID-PASSWORD</td>
<td>4015</td>
</tr>
<tr>
<td>INVALID-PRIORITY</td>
<td>4078</td>
</tr>
<tr>
<td>INVALID-REC-SEQ-NUM</td>
<td>4058</td>
</tr>
<tr>
<td>INVALID-REC-TYPE</td>
<td>4046</td>
</tr>
<tr>
<td>INVALID-RECIP-TYPE</td>
<td>4065</td>
</tr>
<tr>
<td>INVALID-REL-POSITION</td>
<td>4057</td>
</tr>
<tr>
<td>INVALID-REL-TIME-QTY</td>
<td>4074</td>
</tr>
<tr>
<td>INVALID-REQUEST-CODE</td>
<td>8</td>
</tr>
<tr>
<td>INVALID-SESSION-ID</td>
<td>2</td>
</tr>
<tr>
<td>INVALID-UOW-HDR</td>
<td>4</td>
</tr>
<tr>
<td>INVALID-VERSION-CODE</td>
<td>1</td>
</tr>
<tr>
<td>INVALID-WAITPRIORITY</td>
<td>6003</td>
</tr>
<tr>
<td>INVALID-WAITTIMEOUT</td>
<td>6004</td>
</tr>
<tr>
<td>INVALID-ZONE-OFFSET</td>
<td>4021</td>
</tr>
<tr>
<td>IO-TIMEOUT</td>
<td>4990</td>
</tr>
<tr>
<td>ITEM-NOT-FOUND</td>
<td>4035</td>
</tr>
<tr>
<td>ITEM-NOT-IN-FOLDER</td>
<td>4061</td>
</tr>
<tr>
<td>ITEM-NOT-PKG-HDR</td>
<td>4042</td>
</tr>
<tr>
<td>ITEM-TOO-COMPLEX</td>
<td>4036</td>
</tr>
<tr>
<td>ITEM-UNALTERABLE</td>
<td>4041</td>
</tr>
<tr>
<td>ITEMS-EXIST</td>
<td>4213</td>
</tr>
<tr>
<td>ITEMS-IN-FLD</td>
<td>4267</td>
</tr>
<tr>
<td>LIFESPAN-TOO-LONG</td>
<td>4077</td>
</tr>
<tr>
<td>LOGON-DISALLOWED</td>
<td>4007</td>
</tr>
<tr>
<td>MAX-KEY-LEN-ADJUSTED</td>
<td>4101</td>
</tr>
<tr>
<td>MAIL-CTL-REC-ABSENT</td>
<td>4242</td>
</tr>
<tr>
<td>MAIL-PROFILE-ABSENT</td>
<td>4239</td>
</tr>
<tr>
<td>MBR-ALREADY-EXISTS</td>
<td>5662</td>
</tr>
<tr>
<td>MBR-AMBIGUOUS-NAME</td>
<td>5673</td>
</tr>
<tr>
<td>MBR-BAD-NAME</td>
<td>5652</td>
</tr>
<tr>
<td>MBR-BAD-SUFFIX</td>
<td>5674</td>
</tr>
<tr>
<td>MBR-BAD-TYPE</td>
<td>5653</td>
</tr>
<tr>
<td>MBR-NET-DOWN</td>
<td>5661</td>
</tr>
<tr>
<td>MBR-NO-SUCH-NODE</td>
<td>5654</td>
</tr>
<tr>
<td>MBR-NO-PARENT</td>
<td>5658</td>
</tr>
<tr>
<td>MBR-NOT-EMPTY</td>
<td>5659</td>
</tr>
<tr>
<td>MBR-NOT-FOUND</td>
<td>5651</td>
</tr>
<tr>
<td>Text</td>
<td>Error No.</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>MBR-NSRV-DOWN</td>
<td>5657</td>
</tr>
<tr>
<td>MBR-NSRV-ERR</td>
<td>5650</td>
</tr>
<tr>
<td>MBR-NSRV-NOT-FOUND</td>
<td>5656</td>
</tr>
<tr>
<td>MBR-SECURITY</td>
<td>5655</td>
</tr>
<tr>
<td>MEMBERS-EXIST</td>
<td>4225</td>
</tr>
<tr>
<td>MODEL-ALREADY-EXISTS</td>
<td>5737</td>
</tr>
<tr>
<td>MODEL-AMBIGUOUS-NAME</td>
<td>5748</td>
</tr>
<tr>
<td>MODEL-BAD-NAME</td>
<td>5727</td>
</tr>
<tr>
<td>MODEL-BAD-TYPE</td>
<td>5728</td>
</tr>
<tr>
<td>MODEL-DEPOT-ABSENT</td>
<td>4205</td>
</tr>
<tr>
<td>MODEL-NET-DOWN</td>
<td>5736</td>
</tr>
<tr>
<td>MODEL-NO-PARENT</td>
<td>5733</td>
</tr>
<tr>
<td>MODEL-NO-SUCH-NODE</td>
<td>5729</td>
</tr>
<tr>
<td>MODEL-NOT-EMPTY</td>
<td>5734</td>
</tr>
<tr>
<td>MODEL-NOT-FOUND</td>
<td>5726</td>
</tr>
<tr>
<td>MODEL-NOT-SAME-NODE</td>
<td>5747</td>
</tr>
<tr>
<td>MODEL-NSRV-DOWN</td>
<td>5732</td>
</tr>
<tr>
<td>MODEL-NSRV-ERR</td>
<td>5725</td>
</tr>
<tr>
<td>MODEL-NSRV-NOT-FOUND</td>
<td>5731</td>
</tr>
<tr>
<td>MODEL-SECURITY</td>
<td>5730</td>
</tr>
<tr>
<td>MUST-BE-DISC</td>
<td>4392</td>
</tr>
<tr>
<td>MUST-BE-EA</td>
<td>4229</td>
</tr>
<tr>
<td>MUST-BE-IS</td>
<td>4391</td>
</tr>
<tr>
<td>MUST-BE-IWD</td>
<td>4230</td>
</tr>
<tr>
<td>MUST-BE-MD</td>
<td>4384</td>
</tr>
<tr>
<td>MUST-BE-SLT</td>
<td>4382</td>
</tr>
<tr>
<td>MUST-BE-TCN</td>
<td>4383</td>
</tr>
<tr>
<td>MUST-BE-YN</td>
<td>4051</td>
</tr>
<tr>
<td>NEXT-REC-READ</td>
<td>4221</td>
</tr>
<tr>
<td>NO-DEPOT-FOR-CORR</td>
<td>4013</td>
</tr>
<tr>
<td>NO-RECIPS</td>
<td>4082</td>
</tr>
<tr>
<td>NODE-NAME-UNKNOWN</td>
<td>4069</td>
</tr>
<tr>
<td>NOT-CREATED-BY-YOU</td>
<td>4083</td>
</tr>
<tr>
<td>NOT-PROCESSED-HERE</td>
<td>4600</td>
</tr>
<tr>
<td>NOT-UPDATABLE</td>
<td>4302</td>
</tr>
<tr>
<td>ORD-KEY-TRUNCATED</td>
<td>4100</td>
</tr>
<tr>
<td>PARENT-NOT-FOUND</td>
<td>4037</td>
</tr>
<tr>
<td>PASSWORD-MATCH-FAIL</td>
<td>4300</td>
</tr>
<tr>
<td>PAST-DATE-TIME</td>
<td>4071</td>
</tr>
<tr>
<td>PKG-CANCELED</td>
<td>4094</td>
</tr>
<tr>
<td>PKG-EXPIRED</td>
<td>4095</td>
</tr>
<tr>
<td>PKG-NOT-RECEIVED</td>
<td>4080</td>
</tr>
<tr>
<td>Text</td>
<td>Error No.</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>PKG-NOT-SUBMITTED</td>
<td>4084</td>
</tr>
<tr>
<td>PREVIOUSLY-CANCELED</td>
<td>4044</td>
</tr>
<tr>
<td>PREVIOUSLY-SUBMITTED</td>
<td>4043</td>
</tr>
<tr>
<td>PRIORITY-REDUCED</td>
<td>4079</td>
</tr>
<tr>
<td>PRIORITY-SEQUENCE</td>
<td>4310</td>
</tr>
<tr>
<td>PRIV-MUST-BE-RWN</td>
<td>4388</td>
</tr>
<tr>
<td>QUEUE-EMPTY</td>
<td>6001</td>
</tr>
<tr>
<td>REC-ALREADY-EXISTS</td>
<td>4049</td>
</tr>
<tr>
<td>REC-NOT-FOUND</td>
<td>4047</td>
</tr>
<tr>
<td>RECIP-ALREADY-EXISTS</td>
<td>5762</td>
</tr>
<tr>
<td>RECIP-AMBIGUOUS-NAME</td>
<td>5773</td>
</tr>
<tr>
<td>RECIP-BAD-NAME</td>
<td>5752</td>
</tr>
<tr>
<td>RECIP-BAD-SUFFIX</td>
<td>5774</td>
</tr>
<tr>
<td>RECIP-BAD-TYPE</td>
<td>5753</td>
</tr>
<tr>
<td>RECIP-NET-DOWN</td>
<td>5761</td>
</tr>
<tr>
<td>RECIP-NO-PARENT</td>
<td>5758</td>
</tr>
<tr>
<td>RECIP-NO-SUCH-NODE</td>
<td>5754</td>
</tr>
<tr>
<td>RECIP-NOT-EMPTY</td>
<td>5759</td>
</tr>
<tr>
<td>RECIP-NOT-FOUND</td>
<td>5751</td>
</tr>
<tr>
<td>RECIP-NSRV-DOWN</td>
<td>5757</td>
</tr>
<tr>
<td>RECIP-NSRV-ERR</td>
<td>5750</td>
</tr>
<tr>
<td>RECIP-NSRV-NOT-FOUND</td>
<td>5756</td>
</tr>
<tr>
<td>RECIP-SECURITY</td>
<td>5755</td>
</tr>
<tr>
<td>REMOTE-NAME-ACCEPTED</td>
<td>4067</td>
</tr>
<tr>
<td>REPLY-TOO-LONG</td>
<td>6</td>
</tr>
<tr>
<td>RESERVED-MUST-BE-N</td>
<td>4052</td>
</tr>
<tr>
<td>RESTRICTED-OPERATION</td>
<td>4017</td>
</tr>
<tr>
<td>RQST-ERR</td>
<td>3</td>
</tr>
<tr>
<td>RQST-TOO-SHORT</td>
<td>7</td>
</tr>
<tr>
<td>SECURITY-VIOLATION</td>
<td>4093</td>
</tr>
<tr>
<td>SERVICE-DENIED</td>
<td>3</td>
</tr>
<tr>
<td>SESSIONS-ACTIVE</td>
<td>4219</td>
</tr>
<tr>
<td>SPECIAL-FLD</td>
<td>4268</td>
</tr>
<tr>
<td>TFER-CTL-REC-ABSENT</td>
<td>4241</td>
</tr>
<tr>
<td>TFER-PROFILE-ABSENT</td>
<td>4238</td>
</tr>
<tr>
<td>TIME-WINDOW-EXTENDED</td>
<td>4075</td>
</tr>
<tr>
<td>TOO-LATE-TO-DELIV</td>
<td>4608</td>
</tr>
<tr>
<td>TOO-LATE-TO-XPORT</td>
<td>4603</td>
</tr>
<tr>
<td>TOO-MANY-WAITERS</td>
<td>6002</td>
</tr>
<tr>
<td>TSCHED-UNAVAIL</td>
<td>4045</td>
</tr>
</tbody>
</table>
Table A-3. Alphabetic Listing of Error Codes (Continued)

<table>
<thead>
<tr>
<th>Text</th>
<th>Error No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNITS-MUST-BE-DHM</td>
<td>4072</td>
</tr>
<tr>
<td>UOW-NOT-IMPLEMENTED</td>
<td>4003</td>
</tr>
<tr>
<td>UOWS-WITH-ERR</td>
<td>2</td>
</tr>
<tr>
<td>UOWS-WITH-WARNING</td>
<td>1</td>
</tr>
<tr>
<td>UPDATE-MISMATCH</td>
<td>4050</td>
</tr>
<tr>
<td>WAITMANAGER-UNAVAIL</td>
<td>6006</td>
</tr>
<tr>
<td>WAIT-TIMEOUT</td>
<td>6005</td>
</tr>
<tr>
<td>WINDOW-TOO-LARGE</td>
<td>4303</td>
</tr>
<tr>
<td>WRITES-DISALLOWED</td>
<td>4009</td>
</tr>
<tr>
<td>XPORT-FAILED</td>
<td>4602</td>
</tr>
<tr>
<td>XPORTED</td>
<td>4601</td>
</tr>
</tbody>
</table>
APPENDIX B

INTERFACING APPLICATIONS WITH TRANSFER/MAIL

In certain cases, your application might need to interact with the TRANSFER/MAIL (T/MAIL) application supplied by Tandem. This would be necessary, for example, when T/MAIL is embedded within an application, or when you write application software that is called by T/MAIL.

INVOKING T/MAIL FROM YOUR APPLICATION

Your application can invoke T/MAIL through three programmatic entry points:

• MAIL-TERMINAL
• MAIL-PROGRAM
• LOGON-MAIN-A00 USING ipc-hdr

MAIL-TERMINAL is a program unit that is used when T/MAIL is configured as an application program that runs exclusively on a terminal. Once this program unit is invoked, it never exits. The program unit can only be stopped by entering the PATHWAY command STOP TERM through the terminal.

MAIL-PROGRAM is a program unit that exits when the T/MAIL Logon Screen program unit returns as a result of someone pressing the RETURN or EXIT keys. This function is required by the RUN program capability of PATHCOM.

LOGON-MAIN-A00 is a program unit that enables you to embed T/MAIL within another application. The format of IPC-HDR, referenced in the USING clause, is described in Section 4. If the SESSION-ID field in the IPC-HDR structure is not set to the figurative constant LOW-VALUES, T/MAIL bypasses the Logon Screen and displays the Main Menu Screen directly.
Interfacing with T/MAIL

If T/MAIL is invoked and cannot obtain its required configuration of servers, T/MAIL performs an EXIT PROGRAM WITH ERROR operation. The values of TERMINATION-STATUS and TERMINATION-SUBSTATUS will indicate the specific configuration error.

TERMINATION-STATUS = 253: error in GET-CONFIG-NAME UOW
TERMINATION-SUBSTATUS = RETN-CODE of response

TERMINATION-STATUS = 254: SEND ON ERROR for IPC with all GET-CONFIG-NAME UOWs
TERMINATION-SUBSTATUS = TERMINATION-STATUS from SEND ON ERROR

TERMINATION-STATUS = 255: RQST-ERR for IPC with all GET-CONFIG-NAME UOWs
TERMINATION-SUBSTATUS = IPC-RETN-CODE

INVOKING APPLICATION SOFTWARE FROM T/MAIL

You can extend the features supplied by T/MAIL to handle specialized output; for example, displaying mail in a special format or copying mail to some device other than a terminal. To handle this type of task, you must supply your own application software, known in this case as a user mail extension and identified by the program unit name MAIL-EXTENSION-A00.

The person running the application invokes the user mail extension by pressing the shifted F11 (MAIL EXTN) function key on the terminal when any of the following screens are displayed: Scan, Scan Menu, Create Mail, Create Mail Menu, Show Enclosures, or Show Enclosures Menu. This action causes T/MAIL to call your user mail extension, using the following calling convention:

CALL MAIL-EXTENSION-A00
    USING ipc-hdr, lnksd-linkage-block,
    workspace-item, component-item

The IPC-HDR, LNKSD-LINKAGE-BLOCK, WORKSPACE-ITEM, and COMPONENT-ITEM parameters must be declared in the LINKAGE Section of your program unit code in the order presented here.

IPC-HDR indicates the IPC header.

LNKSD-LINKAGE-BLOCK contains general linkage information used between ADMIN and your module. Refer to Section 6 of this guide, and to the TRANSFER Delivery System Management and Administration Guide for details.
INTERFACING WITH T/MAIL

WORKSPACE-ITEM is the T/MAIL workspace used for the construction of new packages and items.

COMPONENT-ITEM indicates the ID of a component item to be displayed or added to the package.

Within your application program, IPC-HDR has the format shown in Section 4. WORKSPACE-ITEM and COMPONENT-ITEM are declared in the following formats.

```
DEF workspace-item.
  02 dummy       PIC X(12).
END.
```

```
DEF component-item.
  02 dummy       PIC X(12).
END.
```

The IPC-HDR structure contains a valid SESSION-ID, allowing your application-supplied program unit or units to request TRANSFER services on behalf of the user. WORKSPACE-ITEM and COMPONENT-ITEM can each contain either an item ID or LOW-VALUES (binary zeros) in specific combinations, with the results noted in Table B-1.

BUILDING PACKAGES THAT T/MAIL CAN READ

When your application prepares packages that are passed to T/MAIL, the packages must be assembled in a format that T/MAIL can read. In planning these packages, the information in the following paragraphs must be considered.

Application ID

When T/MAIL logs on, it uses an application ID of 111. This value is assigned in the APPLIC-ID field of the START-SESSION UOW.

The following APPLIC-ID values are reserved for Tandem use:

- 100-499 Tandem clients; ID 100 indicates a TAREQ, and ID 111 is T/MAIL.
- 500-999 Tandem agents; ID 500 is the VACATION agent.
Interfacing with T/Mail

Table B-1. Effect of WORKSPACE-ITEM and COMPONENT-ITEM Settings

<table>
<thead>
<tr>
<th>WORKSPACE-ITEM CONTENT</th>
<th>COMPONENT-ITEM CONTENT</th>
<th>RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOW-VALUES</td>
<td>item ID</td>
<td>This operation is requested from the Scan Screen, or from the Show Enclosures Screen for an item that is not part of the workspace. Your program unit should display the item indicated by item ID to the user. The program unit can allow the user to modify the item; the program unit, however, does not have enough information to move the item to another folder, or to determine the package to which the item belongs.</td>
</tr>
<tr>
<td>item ID</td>
<td>LOW-VALUES</td>
<td>Your program unit should create a new component item, and add that item to the component items list.</td>
</tr>
<tr>
<td>item ID</td>
<td>item ID</td>
<td>Your program unit can either create a new component and add it to the component list as noted in the above option, or modify the specified component. One situation where both WORKSPACE-ITEM and COMPONENT-ITEM contain item IDs occurs, for instance, when the user enters text on the Create Mail Screen before pressing the shifted F11 key.</td>
</tr>
</tbody>
</table>
Interfacing with T/MAIL

Item Types

T/MAIL handles five types of items, as designated by the ITEM-TYPE field of the item descriptor:

<table>
<thead>
<tr>
<th>Item Type</th>
<th>Value in ITEM-TYPE Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original package</td>
<td>109</td>
</tr>
<tr>
<td>Reply package</td>
<td>110</td>
</tr>
<tr>
<td>Forward package</td>
<td>111</td>
</tr>
<tr>
<td>Unformatted text item</td>
<td>120</td>
</tr>
<tr>
<td>TTEXT item</td>
<td>121</td>
</tr>
</tbody>
</table>

T/MAIL always regards ITEM-TYPE values 109, 110, and 111 as indicating a package.

When T/MAIL presents an item with ITEM-TYPE 120 or 121, it reads all of the records included in the item, independent of their record type. For ITEM-TYPE 120, all of the records must contain only printable characters.

TTEXT items can have embedded ASCII control characters, which are inserted and used for formatting purposes by the T-TEXT text editor. Before displaying a TTEXT item, T/MAIL strips out all embedded ASCII control characters.

Packages and Items

For any package, your application should set the IS-PKG-HDR flag in the ITEM-DESCR-FLAG structure to Y. Then, if the ITEM-TYPE is any value other than 109, 110, or 111, T/MAIL will attempt to treat it as an original package (ITEM-TYPE 109) by default.

Whenever T/MAIL receives a package or an item with an ITEM-TYPE field that does not contain 109, 110, or 111, or in which the item descriptor does not indicate a package header, T/MAIL does not attempt to display the entire package or item; instead, T/MAIL presents only selected data from the item descriptor such as item type, creator name, or creation date.
Interfacing with T/Mail

Record Types

T/Mail handles four types of data records, as designated by the REC-TYPE field in each record:

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Value in REC-TYPE Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject text</td>
<td>115</td>
</tr>
<tr>
<td>Unformatted text</td>
<td>120</td>
</tr>
<tr>
<td>Cc-recipients</td>
<td>323</td>
</tr>
<tr>
<td>To-recipients</td>
<td>340</td>
</tr>
</tbody>
</table>

If a user provides text for the subject field on the Create Mail screen, T/Mail inserts within a package header a subject text record with a REC-TYPE of 115 and a sequence number (REC-SEQ-NUM) of 1 within that record type. When T/Mail reads a package, it searches for the subject text record as a record with REC-TYPE 115 and REC-SEQ-NUM 1. If such a record is not present, T/Mail selects the record with the lowest REC-SEQ-NUM having a REC-TYPE of 115.

The subject text record contains up to 140 bytes of ASCII text. This text must consist of displayable characters only. When T/Mail reads this record, it requests no more than 140 bytes. If the record contains fewer bytes, it is padded with blanks to 140 bytes; if the record contains more than 140 bytes, the excess bytes are truncated from the record. T/Mail always displays the 140-byte record as two 70-byte lines.

The unformatted text record can contain up to 2000 bytes of ASCII text, but T/Mail displays only the first 79 bytes. T/Mail expects unformatted text records (REC-TYPE 120) either in the package header or in a separate item identified by either ITEM-TYPE 120 or ITEM-TYPE 121. Any item created by T/Mail with ITEM-TYPE 120 or 121 contains only records with REC-TYPE 120. T/Mail displays records with REC-TYPE 120 from the package header.

For displaying the recipients of a package, T/Mail depends on the records in the package header with REC-TYPE values of 340 and 323, indicating To-recipients and Cc-recipients, respectively. These records are each 70 bytes long and contain ASCII text, including lists of recipient names separated by commas. The recipient names are resolved as fully as possible at the time the package is submitted, and are stored in that form. For records of REC-TYPE 340 and 323, the REC-SEQ-NUM values must start with 1 and must then increase by 1 for each additional record.
Interfacing with T/MAIL

Recipients

T/MAIL recognizes two types of recipients, as indicated in the RECIP-TYPE field:

<table>
<thead>
<tr>
<th>Recipient Type</th>
<th>Value in RECIP-TYPE Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Courtesy-copy recipient</td>
<td>323</td>
</tr>
<tr>
<td>To recipient</td>
<td>340</td>
</tr>
</tbody>
</table>

T/MAIL adds recipients using these types but does not depend on the TRANSFER recipient lists (available through the GET-RECIP-REC UOW) to display the recipients. To remain compatible with T/MAIL and other applications compatible with T/MAIL, all applications should use these RECIP-TYPE values if possible.

Component Types

T/MAIL uses two values for the COMPNT-TYPE field:

<table>
<thead>
<tr>
<th>Component Type</th>
<th>Value in COMPNT-TYPE Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Undefined/Default</td>
<td>0</td>
</tr>
<tr>
<td>In-reference-to-package</td>
<td>100</td>
</tr>
</tbody>
</table>

T/MAIL searches the component list of a package with ITEM-TYPE 110 or 111 for a component with COMPNT-TYPE 100. This entry indicates the package being replied to or forwarded.

Acknowledging Receipt

Some applications must receive certification of package delivery. After reading the INBOX folder, T/MAIL calls the ACK-RECEIPT UOW under these conditions:

- When the Scan Screen is displayed and a user requests any function that specifically references a single package (such as discard, read, or file a package), T/MAIL acknowledges receipt of the package and then attempts the requested operation.
Interfacing with T/MAIL

- When a user attempts to read the entire INBOX without using the SCAN operation, T/MAIL acknowledges receipt of each package before displaying it.
- When the Detailed Selection Screen is displayed and a user requests the printing of all items in INBOX, T/MAIL constructs the list of items, acknowledges receipt of all items, and then prints the entire list.

Copy Mail Function

For the COPY MAIL function, T/MAIL reports an error if the package header contains more than 13 text records. This can occur if another application sends packages containing more text records. At present, T/MAIL only places up to 13 text records in a package header.

When copying information from a package into the workspace, T/MAIL does not copy data records to the new package. Instead, T/MAIL only copies the component list for the package. Thus, if an application places additional records in the package header, T/MAIL does not preserve those records in the newly made package.

Examples of Standard Packages

Examples of standard send and reply packages appear in Figure B-1. In these examples, a correspondent named SMITH transmits a send package to a recipient named JONES, who responds by transmitting a reply package to SMITH. Each package consists of a package header item and one component item, and references its own recipient list. Each component item contains two or three data records.

The data records in a package can have the same record type (340 or 323) as records in a recipient list, but these records should never be confused because they serve entirely different purposes. Recipient records typically contain delivery information that data records do not; they specify delivery status flags, such as DELIVERED or EXAMINED, as well as other data.
Figure B-1. Standard Send and Reply Packages
APPENDIX C

PROCESSES RUNNING OUTSIDE PATHWAY

PATHWAY is required for configuring a TRANSFER system. As a minimum, this involves the following:

1. Defining the server classes used by TRANSFER and the application; this includes at least one copy of TISERV for use by the TRANSFER TAREQs.

2. Defining the TAREQ SCREEN COBOL object file.

3. Defining the TRANSFER name server that handles the name directory and the TRANSFER scheduler server that handles asynchronous processing; these must be defined as server classes with only one member each.

Once the TRANSFER name and scheduler servers are running, you can start other TRANSFER processes that run outside the PATHWAY operating environment.

Figure C-1 illustrates an application configured to have:

- a requester communicating with its own unique copy of TISERV
- a requester not controlled by PATHWAY, but communicating with a server that runs under PATHWAY.

TRANSFER servers running outside the PATHWAY operating environment must communicate with the TRANSFER name server in order to locate required TRANSFER objects. To permit this communication, the PARAM NAMESPACE must be passed to these TRANSFER servers at startup time.

Processes outside the PATHWAY environment cannot take advantage of PATHWAY load-balancing features, provisions for fault-tolerant operation, and interfacing with TMF. For example:

If a requester not controlled by PATHWAY is communicating with a PATHWAY server and the server cannot adequately handle this
Processes Running Outside PATHWAY

additional requester, the ability of PATHWAY to maintain the system load balance could be affected.

If a server outside the PATHWAY operating environment is not running as a fault-tolerant process and the CPU on which it is running becomes unavailable, you must recreate the server.

If a requester not controlled by PATHWAY requires TMF transaction restart, the requester must handle the restart context management that is normally handled by the PATHWAY RESTART-TRANSACTION verb.

Figure C-1. Application with Processes Outside PATHWAY
When certain events occur, the TAREQ generates TAREQ event packages. The application ID is always TAREQ (100). The agent selector value and the event that triggered the package are listed in Table D-1.

<table>
<thead>
<tr>
<th>Agent-Selector Value</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TRANSFER SYSTEM ERROR</td>
</tr>
<tr>
<td>2</td>
<td>RECIPIENT HAS EXAMINATED CERTIFIED PACKAGE</td>
</tr>
<tr>
<td>3</td>
<td>PACKAGE HAS INVALID RECIPIENT</td>
</tr>
<tr>
<td>4</td>
<td>PACKAGE HAS DISTRIBUTION LIST WITH INVALID RECIPIENT</td>
</tr>
<tr>
<td>5</td>
<td>PACKAGE HAS RECIPIENT WITH INVALID AGENT</td>
</tr>
<tr>
<td>6</td>
<td>PACKAGE COULD NOT BE DELIVERED TO RECIPIENT IN TIME</td>
</tr>
<tr>
<td>7</td>
<td>RECIPIENT DID NOT EXAMINE PACKAGE BEFORE IT EXPIRED</td>
</tr>
<tr>
<td>8</td>
<td>RECIPIENT HAS ALREADY EXAMINED CANCELED PACKAGE</td>
</tr>
<tr>
<td>9</td>
<td>SENDER CANCELED PACKAGE AFTER YOU EXAMINED IT</td>
</tr>
<tr>
<td>10</td>
<td>PACKAGE CANNOT BE TRANSPORTED TO RECIPIENT'S SYSTEM</td>
</tr>
<tr>
<td>11</td>
<td>AGENT IS MISBEHAVING</td>
</tr>
<tr>
<td>12</td>
<td>AGENT HAS LOGGED AN ERROR</td>
</tr>
</tbody>
</table>
APPENDIX E

GLOSSARY

ADMIN application - A TRANSFER application that provides a user interface to the TRANSFER administrative functions; supplied by Tandem.

Agent - A SCREEN COBOL program or PATHWAY server class that is automatically invoked to handle packages received at a depot. Multiple agents can be assigned to a depot; the same agent can be assigned to more than one depot.

Agent profile - A depot profile that defines the criteria for selecting an agent when packages are delivered to the depot. The profile can also contain information that is passed to the agent when the agent is invoked.

Agent selection criteria - Two ranges of values that determine whether or not the agent is invoked. The applicable fields in the package header of an incoming package must be in the ranges indicated by the agent profile in order for the agent to be invoked.

Audited file - A data file that is flagged for auditing by the Transaction Monitoring Facility (TMF); auditing is the monitoring of transactions in preparation for recovery efforts.

Client - A requester program that provides the interface between correspondents and TRANSFER. The client directs requests to the TRANSFER interactive server via the TRANSFER IPC interface. Every TRANSFER application requires at least one client.

Consumer - The process that retrieves an entry from a queue for processing.

Correspondent - A sender and/or receiver of information within the TRANSFER system. A correspondent can be a person, a process, or a device.
Glossary

Correspondent profile - A depot profile that describes the correspondent to TRANSFER. The profile provides default values for package delivery parameters and operational parameters used when the correspondent has a session in progress.

DDL - The Data Definition Language that is used to describe the records and files comprising a data base and the formats of interprocess messages.

Default model depot - The depot whose profiles are copied when a model depot is not specified for a new correspondent. The default model depot is established when TRANSFER is initialized.

Delivery - The TRANSFER action of placing a package in the INBOX folder of a recipient.

Delivery window - The time period between earliest and latest possible delivery for a package.

Depot - That portion of the TRANSFER data base associated with a particular correspondent. A depot includes the profiles, folders, and distribution lists that belong to the correspondent. Each correspondent has exactly one depot, which is created when the correspondent is registered with TRANSFER.

Depot owner - The correspondent for whom the depot was created.

Depot profile record - A profile record that contains information specific to a particular depot. When a new depot is created, all profile records in a model depot are copied into the new depot.

Distribution list - A predefined list of recipient names that provides a simple method for a correspondent to send a package to multiple destinations. A member of a distribution list can be a correspondent or another distribution list. The distribution list belongs to the depot owner, but can be used by any correspondent. Contrast with recipient list.

EMSERV - See Entry Manager.

Entry Manager (EMSERV) - A server that handles enqueuing, reading, and dequeuing of queue entries; an application can have many Entry Managers per Queue file.

Expiration window - The time period between latest possible delivery and scheduled expiration of a package.

Folder - An area in which items and packages can be stored. A folder belongs to the depot owner, and only the depot owner can inspect the contents of the folder.
Fully qualified name - A name that includes the complete name of the correspondent and the network node.

GUARDIAN operating system - The Tandem operating system.

INBOX folder - A special folder that is automatically created for each correspondent. TRANSFER delivers a package to the INBOX folder of each recipient of the package.

Interprocess communication (IPC) - The request to or reply from the TRANSFER interactive server. The IPC contains a header and one or more units-of-work.

IPC header - The first entry in an IPC. The header in an IPC request establishes processing conditions. The header in an IPC reply indicates what occurred during processing.

IPC interface - The means by which a client communicates with TRANSFER.

Item - A collection of information that has a unique identity and that consists of an item descriptor and zero or more data records.

Item descriptor - A record or collection of records that describe the attributes and composition of an item.

Member - An entity named in a distribution list. The member name can be the name of a correspondent or the name of another distribution list.

Model depot - An existing depot whose profiles are copied for a new correspondent.

Name directory - See TRANSFER name directory.

Name resolution - The means by which TRANSFER establishes the validity of a name. TRANSFER fully qualifies the name and then validates the name by checking the TRANSFER name directory.

Name server - A TRANSFER process that manages the TRANSFER name directory.

Node - A Tandem computer system that participates in an EXPAND network.

Object - A generic term used to reference entities managed by TRANSFER. Objects can be: correspondents, depots, distribution lists, folders, items, packages, and profiles.

Ordering criteria - The ordering discipline of a folder, plus the ascending/descending and duplicates options.
Glossary

Ordering discipline - The information TISERV uses as the ordering key when saving items in a folder.

OUTBOX folder - A special folder that is automatically created for each correspondent. The OUTBOX folder is not currently used and is reserved.

Package - A collection of information that can be sent from one correspondent to other correspondents. A package contains at least one item.

Package header - An item that indicates the sender, the recipient list, priority, and delivery and expiration timeframes of a package.

Package lifespan - The maximum length of time TRANSFER monitors a package. If a package has an expiration time, the lifespan is the time period from posting to scheduled expiration. If a package does not have an expiration time, the lifespan is the time period from posting to latest possible delivery.

PATHWAY - A transaction processing system that supplies the programs, procedures, and structures necessary to produce user-written applications.

Pattern - A partial name entered when requesting a list of correspondent, folder, or distribution list names. An asterisk indicates that any characters (zero or more) can appear in that position. Only those names that match the pattern are listed. Contrast with wildcard name.

Posting - The submittal of a package for delivery by TRANSFER.

Process - A running program.

Profile - A description of a correspondent. Each depot has a TRANSFER correspondent profile and a T/MAIL profile. Any depot can also include one or more agent profiles and user-supplied profiles.

Queue - A list of queue entries. Functions are provided to enqueue (add) entries, dequeue (retrieve and delete) entries, read entries, and wait for entries to appear.

Queue entry - Application data that must fit in a single record.

Queue file - An audited key-sequenced file in which multiple queues are stored.

Queue management - An operating environment in which a set of queues are managed for two or more programs.
Glossary

Queue manager - A facility that allows queue entries to be supplied and consumed by different requesters; consists of two server types: Entry Manager and Wait Manager.

Recipient - A correspondent to whom a package is delivered.

Recipient list - The names of correspondents and distribution lists that are to receive a particular package. The recipient list is an attribute of the package and applies only to the package for which the recipient list is created. Contrast with distribution list.

Recipient name - A correspondent name or distribution list name that appears in the recipient list for a particular package.

Registration - A function of TRANSFER by which a correspondent is made known to the system; a depot is automatically created for that correspondent during registration.

Requester process - A process that executes application program object code and sends requests to a server; synonymous with requester.

Scheduler - The TRANSFER process that handles asynchronous events; monitors posted packages and assigns them to TRANSFER asynchronous requesters for delivery at the appropriate time.

SCREEN COBOL - A procedural language that is used to define and control terminal displays.

Server class - A grouping of duplicate copies of a single server program; server processes within the class have identical attributes.

Server process - A process that implements application requests and sends replies to the requester; synonymous with server.

Session - The period of time during which a correspondent can submit requests to TRANSFER. A correspondent must specifically initiate and should terminate a session.

Simple name - A 1- to 32-character name used to identify a correspondent, folder, or distribution list. The name can consist of hyphen (-) and underscore (_) characters; letters A-Z (uppercase and lowercase); and digits 0-9.

Suffix - One or more characters enclosed in parentheses and appended to a correspondent name for use by agents assigned to the depot of the recipient. Any characters except commas, single and double quotation marks, and parentheses can appear in a suffix.
Glossary

Supplier - The process that places an entry on a queue.

System administrator - A correspondent who is responsible for managing TRANSFER objects. Each node has at least one system administrator.

System control record - A profile record that contains global parameters for the node. There can be a unique system control record for any particular REC-TYPE/REC-SEQ-NUM for each node at which your TRANSFER system is running. This record survives deletion of depots, and is not copied into new depots when they are created. Use of information in these records to supply default values for corresponding depot control parameters is determined entirely by your application.

System manager - An individual responsible for initializing, monitoring, and controlling TRANSFER at the local node.

TAL - The Tandem Transaction Application Language that is used to write systems software and routines that support transaction-oriented applications.

TAREQ - See TRANSFER asynchronous requester.

Time window - See delivery window and expiration window.

TISERV - See TRANSFER interactive server.

Transaction Monitoring Facility (TMF) - A data management product that maintains the consistency of a data base and provides the tools for data base recovery.

TRANSFER application - A set of processes that communicate with each other and with TRANSFER, and work together to perform a common task.

TRANSFER asynchronous requester (TAREQ) - A collection of SCREEN COBOL programs that handle the actual delivery of a package to a depot. These programs are supplied by Tandem and run within a standard PATHWAY TCP. The asynchronous requester locates the recipients of a package, delivers the package to local depots, and transports the package to remote nodes for delivery by their asynchronous requesters.

TRANSFER data base - A set of internal files that contain information related to local correspondents and packages in transit. The data base is completely controlled by TRANSFER.

TRANSFER delivery system - A high-level software product that supports communications between people, input/output devices, and processes.
Glossary

TRANSFER interactive server (TISERV) - A TRANSFER server that starts and terminates sessions, services item and package handling requests, and handles administrative requests; supplied by Tandem.

TRANSFER/MAIL (T/MAIL) - A TRANSFER electronic mail application; supplied by Tandem.

TRANSFER name directory - A directory of all names of correspondents, distribution lists, and folders defined at the local node. TRANSFER uses this directory whenever a local name is supplied by a correspondent.

Transporting - The TRANSFER action of sending a package to a remote node for local delivery at that node.

T/MAIL - See TRANSFER/MAIL.

T/MAIL profile - A depot profile that contains parameters used during T/MAIL sessions initiated by the correspondent.

Unit-of-work - An elementary operation to be performed by the TRANSFER interactive server. Several units-of-work can appear within an IPC.

In the queue management environment, an elementary operation to be performed by an Entry Manager or the Wait Manager. Several units-of-work can appear within an IPC, but only one unit-of-work can appear for the Wait Manager.

UOW - See unit-of-work.

Wait Manager (WMSERV) - A server that handles waiting for new queue entries; an application can have only one Wait Manager per Queue file.

WASTEBASKET folder - A special folder that is automatically created for each session started by a correspondent. When a session is terminated, the WASTEBASKET folder for that session is deleted.

Wildcard character - An asterisk included in a TRANSFER simple name to indicate that any character or characters can appear in that position. The separator period must be entered when wildcard characters are used in a fully qualified name. Contrast with pattern.

WMSERV - See Wait Manager.
INDEX

Aborting a TMF transaction 6-20
ACK-RECEIPT UOW 5-5
Acknowledge receipt of a package 5-5
Add
  data record to an item 5-7
  members to distribution lists 5-10
  recipient to a recipient list 5-13
ADD-ITEM-REC UOW 5-7
ADD-MEMBER UOW 5-10
ADD-RECIPI UOW 5-13
ADMIN application 2-5
Administrative UOWs 4-23, 4-29
Agent
  COBOL 2-5
  configuring 6-28
  debugging techniques 6-28
  default configuration screen 5-19, 5-78
  delete 5-17, 5-20, 6-28
  description 1-8, 2-4
  design 2-4, 6-22
  FORTRAN 2-5
  insert 5-17, 5-20
  interprocess communication 2-8
  locating program units 6-24
  names 6-25
  notification message 4-34
  profile 1-15
  program unit 6-24
  programming language 6-27
  reading in order 5-78
  running as PATHWAY requester 6-24
  running as PATHWAY server 6-24
  sample 8-1
  SCREEN COBOL 2-5
  selection 4-33
  selection criteria 5-18, 5-76
  TAL 2-5
  tasks 2-4
  UOWs 4-32
  value ranges 5-19, 5-78
Index

with queue management 9-1
AGENT-SELECTOR field 4-19
Alter data elements in profile record 5-33
ALTER-AGENT-SELECT UOW 5-16
ALTER-FOLDER-ORDER UOW 5-21
ALTER-ITEM-DESCR UOW 5-26
ALTER-PROFILE-ELEM UOW
description 5-33
return codes 5-139
storage layout 5-105
Altering folder order 5-21
APPLIC-ID field 4-19
Application
coding 6-13
defining package formats 6-7
development 6-1
dividing tasks 6-7
error handling 6-11
high-level transactions 6-10
low-level implementation 6-11
nesting packages 6-7
queue management 9-1
screen design 6-5
testing 6-13
TMF guidelines 6-10
Array limits
DDL 5-1
reading profile records 5-164, 5-168
writing profile records 5-212
ASSIGN command
queue management 9-36
TISERV 4-4
Attach component item to parent 5-37
ATTACH-COMPNT-A01 UOW 5-37

BOOLEAN field 4-13
Building a package 1-12, 4-18
Cancel a package 5-40
CANCEL-PKG UOW 5-40
Change agent selection criteria 5-16
Client
COBOL 2-4
coding 2-3
debugging 6-21
description 1-8
design 6-15
FORTRAN 2-4
interprocess communication 2-8
sample 7-1
SCREEN COBOL 2-3
TAL 2-4
tasks 2-3
with queue management 9-1

Index-2
COBOL
  agent 2-5
  client 2-4
Codes, file A-4
COMPNT-TYPE field 4-18
Concurrent sessions 5-120
Configured TISERV name 5-81
Consumer 9-2
Copy an item 5-42
COPY-ITEM UOW 5-42
Correspondent
  addressing restrictions 3-4
  description 1-9
  names 3-2
  privileges 2-17
  profile 1-15
  registration 2-12
Create
  depot 5-44
  distribution list 5-46
  folder 5-48, 5-50
  item 5-54
CREATE-DEPOT UOW 5-44
CREATE-DLIST UOW 5-46
CREATE-FOLDER UOW 5-48
CREATE-FOLDER-B00 UOW 5-50
CREATE-ITEM UOW 5-54
DATE-TIME field 4-20
Debugging
  a client 6-21
  an agent 6-28
Delete
  agent 5-17, 5-20, 6-28
  data record 5-63
  depot 5-56
  distribution list 5-58
  folder 5-60
  members from distribution lists 5-65
  profile record 5-68
  queue entry 9-18
  recipient from recipient list 5-70
DELETE-DEPOT UOW 5-56
DELETE-DLIST UOW 5-58
DELETE-FOLDER UOW 5-60
DELETE-ITEM-REC UOW 5-63
DELETE-MEMBER UOW 5-65
DELETE-PROFILE-REC UOW 5-68
DELETE-RECIP UOW 5-70
Depot
  content 1-13
  delete 5-56
  description 1-13
  establishing 1-13
Index

management 2-14
management UOWs 4-23, 4-30
model 5-44
owner 2-17
security 2-17
DEQ UOW 9-17
Dequeue function 9-3
Designing
agents 6-22
applications 6-1
clients 6-15
Detach a component item from parent 5-72
DETACH-COMPNT UOW 5-72
Developing an application 6-1
Distribution list
create 5-46
delete 5-58
description 1-14, 2-15
management UOWs 4-21, 4-32
name 3-7
Element blocks 5-34, 5-103
EMSERV 9-37
END-SESSION UOW 5-74
End-to-end communication 1-3
Ending a session 5-74
ENQ UOW 9-21
Enqueue function 9-4
Entry Manager
limits 9-2
program file name 9-37
startup 9-32
Errors
queue management initialization 9-38
summary A-1
TAREQ delivery 4-36, A-4
UOW 5-4
Establishing
depot 1-13
session 2-13
Event packages 4-37, D-1
Extended transaction processing 1-3, 6-3
File codes A-4
Files provided with TRANSFER 5-3, 9-15
Folder
altering order 5-21
creating 5-48, 5-50
deleting 5-60
descending order considerations 5-23, 5-52
description 1-15
INBOX 1-15, 5-171
management UOWs 4-31
manipulation UOWs 4-22, 4-28
name 3-5
ordering criteria 5-21
ordering discipline 5-22, 5-51
OUTBOX 1-16
removing items 5-204, 5-205
retrieving folder order 5-82
retrieving items 2-15, 5-207
saving items 5-170, 5-173, 5-177
scanning 5-182, 5-185, 5-188
special 1-15
WASTEBASKET
description 1-16
description 5-74
end of session 5-74
saving items 5-171, 5-174
saving items by key 5-178
unsaving items 5-205
FOLDER-NAME field 4-22
FORTRAN
agent 2-5
client 2-4

GCOB 5-3
GDDL 5-3
Get correspondents with active sessions 5-99
GET-AGENT-SELECT UOW 5-75
GET-CONFIG-NAME UOW 5-79
GET-FOLDER-ORDER UOW 5-82
GET-ITEM-COMPNT-A01 UOW 5-86
GET-ITEM-DESCR UOW 5-89
GET-ITEM-REC UOW 5-94
GET-NEXT-SESSION UOW 5-99
GET-PROFILE-ELEM UOW
description 5-101
return codes 5-116
storage layout 5-105
GET-RECIP-REC UOW 5-144
GLNK 5-3
Glossary E-1
GPQCOB 9-15
GPQDDL 9-15
GPQLNK 9-15
Grouping UOWs 6-16, 9-6
GUARDIAN file codes A-4

Header
IPC 4-7, 9-9
package 2-14, 4-18
UOW 5-2, 9-13

HOLD 6-27

IDLESESSIONDELAY 5-201
INBOX folder 1-15, 5-171
Initiating a session 2-9
Inserting an agent 5-17, 5-20
Interfacing
  with queue management 9-8
  with TRANSFER/MAIL B-1
Interprocess communication, see IPC
IPC
  header 4-7, 9-9
  identifying messages logged 5-152
  LOG-THIS-IPC 4-12, 9-13
  messages 5-152
  PW-REPLY-CODE 4-8, 9-10, A-1
  REQUEST-CODE 4-8, 9-10
  RETN-CODE 4-10, 9-11, A-2
  RETN-CODE-DETAIL 4-11, 9-13, A-3
  SESSION-ID 4-11, 9-13
  size 6-18, 9-9
  UOWS-RETURNED 4-12, 9-13
  UOWS-TO-PROCESS 4-11, 9-13
  VERSION-CODE 4-9, 9-11
Item
  content 1-11
  create 5-54
  description 1-10
  descriptor 1-10
  handling UOWs 4-14, 4-25
  identification 2-11
  nesting 6-36
  retaining access 5-43, 5-55
  size 1-11
  tree operation UOWs 4-17, 4-26
ITEM-ID field 4-16
ITEM-KEY field 4-16
ITEM-TYPE field 4-16
Locating program units 6-24
Lock limits of TMF 6-35
Locking
  dequeue function 9-3
  enqueue function 9-4
Long element blocks 5-35, 5-103
Manager
  Entry 9-2, 9-32, 9-37
  Wait 9-2, 9-32, 9-37
Member
  delete 5-65
  description 1-14
  retrieving 5-153
  retrieving first 5-154
Messages
  agent notification 4-34
  IPC 5-152
  queue management 9-37
Model depot 5-44
MUMPS 2-5

Index-6
Index

NALOAD 5-80
Name
agent 6-25
configuration directory 5-79
correspondent 3-2
directory 3-11
distribution list 3-7
folder 3-5
length restrictions 3-10
management 4-24
management UOW 4-33
node designation 3-10
resolution 2-12, 2-13
searching 3-12
Naming conventions 3-1, 3-10
Nesting items 6-36
Network guidelines 6-38
Node
changing the system number 2-16
name referencing 3-10
NOOP UOW 5-151
Nowait transaction processing 1-3
Object
management 2-11
name resolution 2-12, 2-13
relationships 2-16
OUTBOX folder 1-16
Owner, depot 2-17
Package
building 1-12, 4-18
delivery concepts 5-30
delivery time milestones 5-31
delivery to remote node 2-7, 2-14, 6-38
delivery UOWs 4-21, 4-27
description 1-10
expiration 5-31, 5-203
handling UOWs 4-18, 4-26
header 1-12, 2-14, 4-18
identification 2-11
lifespan 5-31
order of arrival 2-4
priority 4-20
receiving UOW 4-22, 4-28
removal from folders 5-6, 5-31
structure 6-36
submit for delivery 5-202
Package header
adding data records 6-38
content 2-14, 4-18
description 1-12
PATHMON 2-2
PATHWAY environment 1-8
Index

Patterns 3-9
PIC 9(4) COMP fields 4-12, 5-4
PRIORITY field 4-20
Privileges
  access to Profile file elements 5-109
  correspondent 2-17
  depot owner 2-17
  queue management 9-37
  system administrator 2-18
Processes
  interaction 2-2
  operating in TRANSFER applications 2-1, 2-3
  queue management 9-1
  running outside PATHWAY C-1
Profile
  agent 1-15
  correspondent 1-15
  UOWs 4-23, 4-31
Profile file
  access privilege 5-109
  data elements
    ALTER-PROFILE-ELEM return codes 5-139
    GET-PROFILE-ELEM return codes 5-116
  description 1-15
  user-supplied records 6-33
Profile record
  alter 5-33
  delete 5-68
  read 5-162, 5-165
  write 5-210
Profile-element-long 5-35, 5-103
Profile-element-short 5-35, 5-103
Programming languages 1-8

Queue file
  description 9-2
  errors 4-11
  reading sequentially 9-26
  record length 9-3
Queue management
  consumer 9-2
  DEQ UOW 9-17
  ENQ UOW 9-21
  errors A-1
  examples 9-6
  functions 9-3
  initialization errors 9-38
  interaction 9-5
  interface 9-8
  operation 9-1
  reading queues sequentially 9-26
  READQ UOW 9-24
  supplier 9-2
  UOW rules 9-9

Index-8
WAITQ UOW 9-29

Read function 9-4
Read user-maintained profile record 5-162, 5-165
READ-NEXT-MEMBER UOW 5-153
READ-NEXT-NAME UOW 5-157
READ-PROF-REC-A02 5-165
READ-PROFILE-REC UOW 5-162
READQ UOW 9-7, 9-24
REC-SEQ-NUM field 4-17, 5-8
REC-TYPE field 4-17, 5-7
RECIP-KEY field 4-21
RECIP-NAME field 4-21
RECIP-TYPE field 4-21
Recipient
  for local copy of package 5-150
    name 3-4
    remote 5-150
Recipient list
  definition UOWs 4-20, 4-27
  delete 5-70
  description 2-15
  referencing first recipient 5-149
Record
  handling UOWs 4-16, 4-26
  keys 5-97
  locking 6-35
  queue file 9-3
  retrieval 5-96
Registration 2-12
Removing
  items from folders 5-204
  packages from folders 1-16, 5-203
  queue entries 9-3, 9-18
Resolving object names
  deferred resolution 2-13
  description 2-12, 3-12
  for distribution lists 2-13
  for recipient lists 2-13
  immediate resolution 2-13
Retain access to new items 5-43
Retrieving
  agent selection criteria 5-76
  component items in a parent 5-86
  configured name 5-79
  data elements from Profile file 5-101
  data records from an item 5-94
  fields from item descriptor 5-89
  folder names 5-207
  folder ordering criteria 5-82
  items from folders 2-15
  members of distribution lists 5-153
  names 5-157
  queue entries 9-17
Index

recipient records 5-144

Sample
agent 8-1
client 7-1
Save an item in a folder 5-170, 5-173, 5-177
SAVE-ITEM UOW 5-170
SAVE-ITEM-B00 UOW 5-173
SAVE-ITEM-BY-KEY UOW 5-177
Scan contents of a folder 5-182, 5-185, 5-188
SCAN-FOLDER UOW 5-182
SCAN-FOLDER-B00 UOW 5-185
SCAN-FOLDER-BY-KEY UOW 5-188
Scheduler 2-7
SCREEN COBOL agent 2-5, 6-24
SCREEN COBOL client 2-3
Screen design 6-5
Security
checking by TRANSFER 2-17
depot 2-17
queue management 9-37
Selecting
agents 4-33
item data records 5-95
Server
Entry Manager 9-2, 9-32, 9-37
queue management 9-32
specialized 2-5
TRANSFER 2-5
TWORK 6-37
user-written 2-5
Wait Manager 9-2, 9-32, 9-37
Session
concurrency 5-120
control UOWs 4-14, 4-25
elements 2-9
ending 5-74
establishing 2-13
for agents 6-26
ID 2-9
idle 5-201
management UOW 4-23, 4-30
starting 2-9, 5-198
terminating 5-74, 5-100
Short element blocks 5-34, 5-103
Simple names 3-1
Software provided with TRANSFER 5-3, 9-15
Special folders 1-15
Specialized servers 2-5
Start
a session 5-198
queue management servers 9-32
TISERV 4-1
START-SESSION UOW 5-198

Index-10
Storing recipient names 3-4
Submit a package for delivery 5-202
SUBMIT-PKG UOW 5-202
Suffix 3-4, 6-23
Supplier 9-2
System administrator
   description 1-15
   privileges 2-18
System control modules 6-34
System control record 5-211

TAL
   agent 2-5
   client 2-4

TAREQ
   agent notification message 4-34
   agent selection 4-33
   calling a SCREEN COBOL agent 4-37
   delivery errors 4-41
   description 2-7
   ERROR-RETURN 4-36
   errors A-4
   event packages 4-37, D-1
   guidelines 6-37
   interface 4-33, 6-26
   precautions 4-41
   program unit 6-24
   queue management 9-6
   sending to a server class agent 4-37
   TMF processing 6-27

TAREQCOD 6-24
TAREQDIR 6-24
Terminate a session 5-74, 5-100
Time zone differences 5-200

TISERV
   configured name 5-81
   functions 2-5, 4-1
   guidelines 6-36
   interface 4-5
   parameters at startup time 4-2
   remote copy 5-150
   reply 4-5
   request 4-5
   starting 4-1

TMF
   description 1-7
   guidelines 6-10, 6-19
   lock limits 6-35, 9-3
   processing 6-34
   queue management 9-3
   record locking 6-35, 9-3
   TAREQ processing 6-27
   TRANSID 2-6
   when to abort a transaction 6-20
Index

Transaction Monitoring Facility, see TMF
Transaction packaging 6-11
TRANSFER
agent 1-8
applications 1-3, 6-1
client 1-8
configuration UOW 4-22, 4-29
error codes A-1
limitations 1-6
name directory 3-11
name searching 3-12
naming conventions 3-1
objects 1-9
overview 1-1
queue management 9-1
screen design 6-5
servers 2-5
software provided 5-3
tasks 1-5
TRANSFER asynchronous requester, see TAREQ
TRANSFER interactive server, see TISERV
TRANSFER/MAIL interface B-1
TRANSID 2-6
Transmit a dummy UOW 5-151
TSCHED 6-22
tWOrK 6-37

Unit-of-work, see UOW
Unsave time
changing 5-181
saving items 5-174, 5-178
scanning folders 5-187, 5-194
UNSSAVE-ITEM UOW 5-204
UOW
administrative 4-23, 4-29
agent 4-32
depot management 4-23, 4-30
descriptions 5-1, 9-15
distribution list management 4-24, 4-32
ELEM-RETN-CODE errors A-4
errors 5-4
filler fields 5-1, 5-8
folder management 4-31
folder manipulation 4-22, 4-28
functional areas 4-12
grouping 6-16, 9-6
header 5-2, 9-9, 9-13
item handling 4-14, 4-25
item tree operations 4-17, 4-26
MBR-RETN-CODE errors A-4
modifying array limits 5-1
name management 4-24, 4-33
package delivery 4-21, 4-27
package handling 4-18, 4-26
package receiving 4-22, 4-28
profile 4-31
queue management 9-16
recipient list 4-20
recipient list definition 4-27
record handling 4-16, 4-26
request 5-2, 9-9, 9-13
response 5-2, 9-9, 9-13
RETN-CODE 5-2, A-2
RETN-CODE-DETAIL 5-3, A-3
session control 4-14, 4-25
session management 4-23, 4-30
submittal preparation 4-18, 4-27
summary 4-25
TMF transactions 5-1
TRANSFER configuration 4-22, 4-29
whole item operations 4-15, 4-25
word boundaries 5-8
UPDATE-CONTROL field 4-13, 5-28
Updating fields in item descriptor 5-26
User-written server 2-5

Version codes 4-9

Wait function 9-4
Wait Manager
    errors 4-11
    limits 9-2
    program name 9-37
    startup 9-32
WAITQ UOW 9-29
WASTEBASKET folder
    description 1-16
    end-of-session 5-74
    saving items 5-171, 5-174
    saving items by key 5-178
    unsaving items 5-205
WHERE-SAVED UOW 5-207
Wildcard feature 3-9
WMSERV 9-37
Word boundaries 5-8
WRITE-PROFILE-REC UOW 5-210
Writing a user-maintained profile record 5-210

XBEGIN 5-80
READER COMMENT CARD

Tandem welcomes your comments on the quality and usefulness of its software documentation. Does this manual serve your needs? If not, how could we improve it? Your comments will be forwarded to the writer for review and action, as appropriate.

If your answer to any of the questions below is “no,” please supply detailed information, including page numbers, under Comments. Use additional sheets if necessary.

► Is this manual technically accurate? 
  Yes ☐ No ☐

► Is information missing? 
  Yes ☐ No ☐

► Are the organization and content clear? 
  Yes ☐ No ☐

► Are the format and packaging convenient? 
  Yes ☐ No ☐

Comments

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

Name

Company

Address

City/State

Zip

TRANSFER™ Delivery System Programming Guide
NonStop™ Systems
NonStop 1™ System
82525 A00
## TRANSFER UOWs

<table>
<thead>
<tr>
<th>Transfer UOW</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACK-RECEIPT</td>
<td>5-5</td>
</tr>
<tr>
<td>ADD-ITEM-REC</td>
<td>5-7</td>
</tr>
<tr>
<td>ADD-MEMBER</td>
<td>5-10</td>
</tr>
<tr>
<td>ADD-RECIP</td>
<td>5-13</td>
</tr>
<tr>
<td>ALTER-AGENT-SELECT</td>
<td>5-16</td>
</tr>
<tr>
<td>ALTER-FOOLDER-ORDER</td>
<td>5-21</td>
</tr>
<tr>
<td>ALTER-ITEM-DESCR</td>
<td>5-26</td>
</tr>
<tr>
<td>ALTER-PROFILE-ELEM</td>
<td>5-33</td>
</tr>
<tr>
<td>ATTACH-COMPNT-A01</td>
<td>5-37</td>
</tr>
<tr>
<td>CANCEL-PKG</td>
<td>5-40</td>
</tr>
<tr>
<td>COPY-ITEM</td>
<td>5-42</td>
</tr>
<tr>
<td>CREATE-DEPOT</td>
<td>5-44</td>
</tr>
<tr>
<td>CREATE-DLIST</td>
<td>5-46</td>
</tr>
<tr>
<td>CREATE-FOOLDER</td>
<td>5-48</td>
</tr>
<tr>
<td>CREATE-FOOLDER-B00</td>
<td>5-50</td>
</tr>
<tr>
<td>CREATE-ITEM</td>
<td>5-54</td>
</tr>
<tr>
<td>DELETE-DEPOT</td>
<td>5-56</td>
</tr>
<tr>
<td>DELETE-DLIST</td>
<td>5-58</td>
</tr>
<tr>
<td>DELETE-FOOLDER</td>
<td>5-60</td>
</tr>
<tr>
<td>DELETE-ITEM-REC</td>
<td>5-63</td>
</tr>
<tr>
<td>DELETE-MEMBER</td>
<td>5-65</td>
</tr>
<tr>
<td>DELETE-PROFILE-REC</td>
<td>5-68</td>
</tr>
<tr>
<td>DELETE-RECIP</td>
<td>5-70</td>
</tr>
<tr>
<td>DETACH-COMPNT</td>
<td>5-72</td>
</tr>
<tr>
<td>END-SESSION</td>
<td>5-74</td>
</tr>
<tr>
<td>GET-AGENT-SELECT</td>
<td>5-75</td>
</tr>
<tr>
<td>GET-CONFIG-NAME</td>
<td>5-79</td>
</tr>
<tr>
<td>GET-FOLDER-ORDER</td>
<td>5-82</td>
</tr>
<tr>
<td>GET-ITEM-COMPNT-A01</td>
<td>5-86</td>
</tr>
<tr>
<td>GET-ITEM-DESCR</td>
<td>5-89</td>
</tr>
<tr>
<td>GET-ITEM-REC</td>
<td>5-94</td>
</tr>
<tr>
<td>GET-NEXT-SESSION</td>
<td>5-99</td>
</tr>
<tr>
<td>GET-NEXT-MEMBER</td>
<td>5-153</td>
</tr>
<tr>
<td>READ-NEXT-NAME</td>
<td>5-157</td>
</tr>
<tr>
<td>READ-NEXT-MEMBER</td>
<td>5-153</td>
</tr>
<tr>
<td>READ-PROFILE-REC</td>
<td>5-162</td>
</tr>
<tr>
<td>READ-PROFILE-REC-A02</td>
<td>5-165</td>
</tr>
<tr>
<td>SAVE-ITEM</td>
<td>5-170</td>
</tr>
<tr>
<td>SAVE-ITEM-B00</td>
<td>5-173</td>
</tr>
<tr>
<td>SAVE-ITEM-BY-KEY</td>
<td>5-177</td>
</tr>
<tr>
<td>SCAN-FOLDER</td>
<td>5-182</td>
</tr>
<tr>
<td>SCAN-FOLDER-B00</td>
<td>5-185</td>
</tr>
<tr>
<td>SCAN-FOLDER-BY-KEY</td>
<td>5-188</td>
</tr>
<tr>
<td>START-SESSION</td>
<td>5-198</td>
</tr>
<tr>
<td>SUBMIT-PKG</td>
<td>5-202</td>
</tr>
<tr>
<td>UNSAVE-ITEM</td>
<td>5-204</td>
</tr>
<tr>
<td>WHERE-SAVED</td>
<td>5-207</td>
</tr>
<tr>
<td>WRITE-PROFILE-REC</td>
<td>5-210</td>
</tr>
</tbody>
</table>

## Queue Management UOWs

<table>
<thead>
<tr>
<th>Queue Management UOW</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEQ</td>
<td>9-17</td>
</tr>
<tr>
<td>ENQ</td>
<td>9-21</td>
</tr>
<tr>
<td>READQ</td>
<td>9-24</td>
</tr>
<tr>
<td>WAITQ</td>
<td>9-29</td>
</tr>
</tbody>
</table>