
Unicenter

Management Services

EIP

User Reference, Planning, and Installation

4th edition

P01- 101



Computer Associates
The Software That Manages eBusiness



Manual History

Edition	Publication Number	Product Version	Min. MS Level	Publish Date
1st Edition	P01-22101-00	Version 2.2	NET/MASTER 2.2	October 1991
2nd Edition	P01-30101-00	Version 3.0	ASM 3.0	July 1993
3rd Edition	P01-101-31	Version 3.1	MS 3.1	May 1994
4th Edition	P01-101	Version 3.2	MS 3.2	March 1995
4th Edition—Update 1	P01-101	Version 3.2	MS 3.2	March 1998

New editions incorporate any updates issued since the previous edition.

SOLVE Management Services™ External Interface Package User Reference, Planning and Installation, 4th Edition, P01-101

This document was prepared to assist licensed users with the use of SOLVE Management Services™; its contents may not be used for any other purpose without prior written permission, and this manual is subject to the terms and conditions of the applicable license agreement. The material contained herein is supplied without representation or warranty of any kind. Sterling Software and its subsidiaries, therefore, assume no responsibility and shall have no liability of any kind arising from the supply or use of this document or the material contained herein.

References in this manual to Sterling Software products, programs, or services do not imply that Sterling Software intends to make these available in all countries in which Sterling Software operates.

All rights reserved, including the right to reproduce this document or any portion thereof in any form. Copyright© 1991, 1998, Sterling Software, Inc.

The status symbols ® and ™, as used to identify Sterling Software trademarks herein, refer to the status of Sterling Software trademarks as pending or registered in the U.S. Patent and Trademark Office as of July 1992. Sterling Software has applied for and has been granted registrations for their trademarks in various countries throughout the world. Sterling Software will act to protect its trademark rights worldwide.

INFO/MASTER is a registered trademark of Sterling Software, Inc. in the U.S.A. and/or other countries. AOM, EASINET, External Interface Package, EIP, Expert System Foundation, ESF, Inter-Management Services Connection/Extended Function, INMC/EF, Multiple Application Interface, MAI, MAI/EF, Network Control Language, NCL, Network Control System, NCS, Network Error Warning System, NEWS, Network Tracking System, NTS, NET/MAIL, SOLVE, SOLVE:Access, SOLVE:Asset, SOLVE:Attach, SOLVE:Automation, SOLVE:Central, SOLVE:Change, SOLVE:Commander, SOLVE:Configuration, SOLVE:Connect, SOLVE:Information, SOLVE:Link, SOLVE:Monitor, SOLVE:Netmaster, SOLVE:Operations, SOLVE:Problem, and SOLVE:Stat are trademarks of Sterling Software, Inc. and or its subsidiaries.

Tell Us What You Think

External Interface Package User Reference, Planning and Installation, 4th Edition

Your appraisal of this book will help us make quality improvements in upcoming releases. Feel free to copy and distribute this page to others on your staff. Please return it to the address shown on the reverse side of this page.

What changes in this book would help you most? Please check no more than three.

- Add more examples.
- Add more information about installing and customising the product.
- Expand information in areas necessary to use or run the product.
- Improve clarity and consistency of writing style.
- Improve organisation of information.
- Improve technical accuracy of the information.
- Include more information about troubleshooting and solving typical problems.
- Update the index or table of contents to make information easier to locate.

Please use this space for any additional comments or suggestions you may have:

Please identify yourself so that we can contact you directly if we have any questions about your remarks:

Your name: _____

Company name: _____

Site/address: _____

Telephone number: _____

Send Us Your Comments

We are very happy to receive your suggestions, comments, or questions that will make SOLVE management services manuals more useful to you and to other users.

Your comments will go directly to the Publications Editor who will review them with the writer and the technical specialist who have primary responsibility for this manual. The Publications Editor will keep you informed of all actions resulting from your comments.

How to Send Us Your Comments

To send us your comments, you can use one of the methods shown below:

Method	Address
Mail	Sterling Software Publications Locked Bag 504 Frenchs Forest, NSW 2086 Australia
Fax	61 2 975 2921 or fax your local Sales or support centre
Internet	editor@sydney.sterling.com
Phone	61 2 975 4777 or phone your local Sales or Support Representative

If you mail your comments, please mark or write them on the pages of the manual where they apply. (Use additional blank pages if necessary.) Then, complete a copy of the Tell Us What You Think page and include it with your comments.

If you fax your comments, use the same method as for mailing your comments, and also complete a copy of the FAX TRANSMITTAL SHEET on the following page.

If you send your comments through Internet, please include the following information:

- ▶ Name and edition of manual
- ▶ Book Code; for example, P01-101
- ▶ Page number and comments that apply to that page

FAX TRANSMITTAL SHEET

P01-101

External Interface Package User Reference, Planning and Installation, 4th Edition

Date: _____

To: Technical Communications

At FAX #: (02) 975 2921

Number of pages, including this cover sheet: _____

From: _____

Title: _____

Company: _____

Address: _____

Telephone Number: _____

Message: _____

Contents

	What's New in This Edition	xiii
	About This Manual	xv
Chapter 1	Introduction	1-1
	The External Interface Package (EIP)	1-1
	EIP Components	1-2
	Interface Types	1-2
	SOLVE Management Services Interfaces	1-3
	Connecting to SOLVE Management Services	1-3
	Signing on to SOLVE Management Services	1-3
	Continuing the Dialogue	1-4
	EIP Operational Environment	1-4
Chapter 2	TSO Full-Screen Interface	2-1
	The NMLOGON Command	2-2
	Examples:	2-3
	Return Codes:	2-3
	Autologon	2-4
	Operating Differences	2-4
	The EASINET Option	2-5
	Exiting from the TSO Full-Screen Interface	2-5
Chapter 3	TSO Command Interface	3-1
	The NMCMD Command	3-2
	Examples:	3-3
	NMCMD Sub-Commands	3-4

	Issuing SOLVE or VTAM Commands	3-5
	The NMCMD Operating Environment	3-5
	TSO Batch Execution of NMCMD	3-6
	Return Codes:	3-6
	Examples:	3-7
	Example 1:	3-7
	Example 2:	3-7
	Example 3:	3-8
	Example 4:	3-8
Chapter 4	Batch Command Interface (BCI)	4-1
	BCI Command Types	4-1
	BCI Input	4-2
	BCI Output	4-3
	The BCILogon Command	4-4
	Examples:	4-5
	The BCIDISC Command	4-6
	The BCIEXITC Command	4-6
	BCI Operation	4-7
	Procedure \$USERBCI	4-7
	Customising \$USERBCI	4-9
	The BCI Command Exit	4-9
	Return Codes:	4-10
	JCL Return Codes	4-11
	Examples:	4-12
	Example 1: VSE System	4-12
	Example 2: OS System	4-12
Chapter 5	Virtual 3270 Interface	5-1
	Using the Virtual 3270 Interface	5-1
	Controlling the Conversation	5-3
	Interfacing to NCL Procedures	5-3
	Panel Characteristics	5-3
Chapter 6	FTS External Interfaces Overview	6-1
	The FTS NCL Interface	6-1
	Support for TSO Users	6-2
	NCL Interface Operational Environment	6-2
	FTS Procedures	6-3
	Transmission Status Information	6-3
	NCL Access to VFS	6-4

Chapter 7	FTS NCL Interface	7-1
	The &FTS Variables	7-1
	NCL Procedure Descriptions	7-7
	Limitations of FTS Procedures	7-10
	Customising NCL Procedures	7-11
	Name Masking	7-11
	Status Display Filtering	7-11
	Dynamic Transmission Definition Generation	7-11
	Private and System Definitions	7-12
Chapter 8	FTS TSO Interface	8-1
	The NMCMD Command Processor	8-1
	The FTS Sub-Command Environment	8-1
	The NMCMD FTSDEF Sub-Command	8-3
	The NMCMD FTSEDEL Sub-command	8-8
	The NMCMD FTSSSEND Sub-command	8-9
	The NMCMD FTSXMIT Sub-command	8-10
	TSO Interface NCL Procedures	8-11
	FTSDEF	8-11
	FTSXMIT	8-12
	FTSEDEL	8-12
	FTSSSEND	8-12
	Messages Returned to TSO	8-13
	Return Codes From \$USTSFTS	8-15
	Examples of FTS Requests Using NMCMD	8-15
	Example 1	8-16
	Example 2	8-16
	Example 3	8-17
	Example 4	8-17
Chapter 9	FTS ISPF Dialogue Interface	9-1
	Functional Overview	9-1
	Direct File Transfers	9-2
	Staged File Transfers	9-2
	Edit Restrictions	9-3
	Dialogue Components	9-3
	Dialogue Procedures	9-3
	NMFTS	9-3
	NMFTSC	9-3
	NMFTSM	9-4
	NMFTSS	9-4
	NMFTSID	9-4
	NMFTSED	9-4

Dialogue Programs	9-4
NMFTDSRG	9-4
NMFTMSEL	9-5
Dialogue Panels	9-6
NMFTFR	9-6
NMFTTO	9-6
NMFTNEW	9-6
NMFTMSEL	9-7
NMFTSMM	9-7
NMFTVSE	9-7
NMFTVSR	9-7
NMFTDSTG	9-7
NMFTWAIT	9-7
NMFTSERR	9-7
NMFTSOK	9-7
NMFTHxxx	9-7
Dialogue Messages	9-8
NCL Procedures	9-8
\$@COPY	9-8
\$@DELETE	9-8
@E	9-8
\$L	9-8
\$@M	9-8
\$@NMFTS	9-9
\$@NOTIFY	9-9
\$@R	9-9
\$@REPRO	9-9
\$@S	9-9
SOLVE Management Services Utilities	9-9
UTIL0007	9-9
UTIL0009	9-9
Tailoring the Dialogue	9-10
Procedure NMFTSID	9-10
SYSID	9-10
LCLPREF	9-10
LCLUNIT	9-10
LCLVOL	9-11
REMPREF	9-11
REMUNIT	9-11
&REMVOL	9-11
VSAMSPC	9-11
FTSTYP	9-11
MSGOPT	9-12
&SYSFIDn, &SYSTIDn, &NMAPPLn and &NMLINKn	9-12
Procedure \$@NMFTS	9-12
PREFIX	9-13
SYSOPER	9-13
Operational Considerations	9-13
Invoking the Dialogue	9-13
FTS Transmission Request Names	9-14

Managing Temporary Datasets	9-14
Error Recovery	9-15
Using the Dialogue	9-16
Appendix A Customer Support Services	A-1
Technical Support	A-1
Local Support Centres	A-2
Europe	A-3
North America	A-4
Pacific Rim	A-5
User Conferences	A-5
Appendix B Installation Considerations	B-1
Installing EIP Components	B-1
EIP Load Libraries	B-1
OS Load Libraries	B-1
VSE Core-Image Libraries	B-2
EIP NCL Procedures	B-2
VTAM APPL Definitions	B-3
SOLVE Management Services User IDs	B-3
The Sample BCI Command Exit	B-4
Testing the Interfaces	B-5
The FTS External Interfaces	B-5
FTS NCL Interface	B-5
Testing the FTS NCL Interface Procedures	B-5
User Services Invocation	B-6
FTS TSO Interface	B-7
Testing the FTS TSO Interface Procedures	B-7
FTS ISPF Dialogue Interface	B-7
The TSO Environment	B-7
Non-TSO/E Considerations	B-8
SOLVE Management Services Environment	B-8
Use of FROM and TO Commands	B-9
Staged PDS Transmission Requirements	B-10
IEBCSOD	B-10
IEBCSILN	B-10
IEBCSILR	B-10
Security Considerations	B-11

Glossary

Index

What's New in This Edition

This manual is the 4th edition of the *External Interface Package User Reference, Planning and Installation* manual (publication number P01-101).

Since the 3rd edition, the only change to this manual is a correction to the NMCMD command description on page 3-2, indicating that the USER operand is optional.

About This Manual

This manual is the principal reference for:

- ▶ Those users who install and customise the External Interface Package procedures and panels, and
- ▶ Those users responsible for planning to use External Interface Package procedures

It is assumed that readers of this manual are already familiar with SOLVE management services and also the File Transmission Services feature (FTS).

For information about these products, see the following manuals:

- ▶ *SOLVE Management Services Planning and Installation*
- ▶ *SOLVE Management Services User's Guide*
- ▶ *FTS Planning and Installation*
- ▶ *FTS User Reference*

Introduction

The External Interface Package (EIP)

EIP is a component of SOLVE management services. It provides an alternative, convenient way of accessing SOLVE management services facilities in certain situations. For example, SOLVE management services access may be required without having to use a terminal, or where a terminal is not able to connect to SOLVE management services directly.

Under normal conditions, a user gains direct access to SOLVE management services by logging onto the system from a terminal in the network. However this may be inconvenient for ad hoc requests, or where access is required for casual users, or by other programs or jobstreams.

EIP lets you connect to SOLVE management services from various external sources, on *behalf* of a known user. Different forms of access are provided depending on the different types of environment involved.

For example, often systems and operational personnel will need rapid access to standard SOLVE management services facilities without logging off from TSO. Or perhaps a batch or on-line program must pass a series of commands to SOLVE management services to initiate or automate a range of external processes. EIP is able to satisfy these requirements.

This chapter describes the various EIP components and their operational characteristics and requirements. It also describes the steps required to customise the EIP feature, where necessary.

EIP Components

EIP includes:

- ▶ Standard Assembler modules
- ▶ TSO command processors (for OS/VS systems)
- ▶ Network Control Language (NCL) procedures and panels
- ▶ Panels, modules, and ISPF Dialogue Manager CLISTS for the external FTS implementation, where required

Interface Types

The various EIP components provide a number of different interface types, each with different operating characteristics. The different interface types are described in separate chapters in this part of the manual, as follows:

- ▶ The *TSO Full-screen Interface* invokes a TSO command processor named NMLOGON, and provides full-screen access to SOLVE management services directly from TSO.
- ▶ The *TSO Command Interface* invokes a TSO command processor named NMCMD, and provides line-by-line access to SOLVE management services. This allows commands to be entered and the results returned.
- ▶ The *Batch Command Interface* is a standard load module named NMBCI that can be executed from a batch environment to enter commands and log the messages produced.
- ▶ The *Virtual 3270 Interface* is a utility load module named NMV3270I that provides services for a pass-through dialogue with SOLVE management services.

Note: The TSO Command Interface and the Batch Command Interface both use the services provided by the Virtual 3270 Interface.

All these forms of access to SOLVE management services through EIP are supported under MVS and MSP operating systems.

Under VSE operating systems, EIP access is restricted to Batch Command and Virtual 3270 Interfaces only.

Note: Choose the access program appropriate to the type of function required. Although servicing different requests, all interfaces have some features and requirements in common. These common features are discussed next.

SOLVE Management Services Interfaces

All interfaces have similar ways of operating. Firstly, a connection must be made to the desired SOLVE management services system, then the user signon is checked to validate the connection before entering the SOLVE management services area which will handle the session.

Connecting to SOLVE Management Services

Connecting to the appropriate SOLVE system requires that the interface module establish a VTAM application to application session with that system. To do this the interface module opens a VTAM ACB and attempts to set up a session as though the request originated from a terminal.

This means that VTAM ACB definitions must be available for the interface module to use. All interfaces are capable of searching for an appropriate definition or of being supplied with the specific name required. Further details on defining default ACB names are discussed in the appendix on *Installation Considerations*; instructions on using specific names can be found in the chapters dealing with each of the relevant interfaces.

By using standard VTAM definitions it is possible to target the session request at SOLVE management services on the same or another network host. Once the session is established, the dialogue proceeds through the interface, depending upon the particular user requests.

Signing on to SOLVE Management Services

Normally when a terminal requests a session with SOLVE management services the user at the terminal must logon by specifying a user ID and password. In order not to compromise the SOLVE management services security, this is generally true of a logon from an external interface.

However, SOLVE management services can detect when a logon comes from an external interface and allows logon to proceed without the need to supply a password, provided it is authorised by the User Access Maintenance System (UAMS), or your installation's equivalent security system. This authorisation procedure is primarily used when targeting

a logon request from another source which is itself deemed to be secure (for example, from TSO where the user has already signed on to TSO).

Otherwise, whenever a request comes from somewhere other than an on-line TSO user, both the user ID and password must be supplied when connecting to SOLVE management services. See the appendix on *Installation Considerations* for more information on user detail requirements, and each interface chapter for details on how the user ID and password are entered.

Continuing the Dialogue

Once the user is signed onto SOLVE management services, the conversation proceeds according to the type of interface being used. The session is driven according to the input passed to the interface, and the disposition of any replies output is determined by the type of interface used.

For the TSO Full-screen Interface this results in a standard SOLVE management services session. For the other interface types, NCL procedures are normally used to service the requests within SOLVE management services and to furnish the results to be returned.

Any commands or messages generated within SOLVE management services during the course of this session are directed to the activity log and accredited to the user in the usual manner. Normal command authority checks and system access privileges apply.

EIP Operational Environment

The specific components and requirements of each EIP function are fully documented in the relevant sections of the manual following.

TSO Full-Screen Interface

The TSO Full-Screen Interface is invoked by using the NMLOGON command from within a TSO session. It is available in OS/MVS or MSP environments only, although it may be used to connect to SOLVE management services running under any operating system.

The interface establishes a *full-screen session* which provides direct entry to the SOLVE management services. With only slight operational differences, all SOLVE management services facilities can be executed from this interface in the usual manner.

The interface will be used to connect to a SOLVE management services operating in a different VTAM domain.

The NMLOGON Command

The TSO Full-Screen Interface operates as a command processor and is invoked by using the NMLOGON command. The format of the NMLOGON command is:

```
NMLOGON [  APPL( applname ) ]  
        [  AUTO | NOAUTO ]  
        [  EASINET | EASINET( data data ... data ) ]  
        [  LU( luname ) ]  
        [  MENU( selection ) ]
```

APPL(*applname*)

Provides the VTAM APPL name of the SOLVE management services you wish to log on to. The default if this operand is omitted is APPL(NM).

AUTO | NOAUTO

Determines whether an *autologon* is to be attempted (see below). The default if this operand is omitted is AUTO.

EASINET | EASINET(*data data ... data*)

Signifies you require access to the facility to target SOLVE management services :

If the EASINET keyword is omitted, it is assumed that you wish to log onto SOLVE management services proper, bypassing any EASINET system that is in place.

If this keyword is supplied and the target SOLVE management services is running EASINET, the TSO session is placed within EASINET. Subsequent processing is then controlled by that EASINET procedure.

The TSO user ID is made available to the EASINET procedure in a user variable called &1. If the NOAUTO keyword is also supplied, &1 will contain the word *UNKNOWN. If data is supplied with the EASINET keyword, it is tokenised into multiple words and made available to EASINET in user variables &2, &3, &4, etc. (this can be used to pass required processing options to EASINET, for example).

LU(*luname*)

Specify the entire ACB name NMLOGON will attempt to open to connect to SOLVE management services. If the open fails for any reason, NMLOGON processing terminates immediately.

LUPREF(*luprefix*)

Specify a 1–5 character prefix for the series of ACB names that NMLOGON will attempt to use when connecting to SOLVE management services. A suffix of 001 is initially added to the *luprefix* supplied, and an attempt made to open the ACB. If this ACB is in use, the suffix keeps incrementing and the open retried until an open is successful, an unrecoverable open error occurs, or the list of defined ACBs is exhausted. If this operand is omitted, a default prefix of NMTSO is used.

MENU(*selection*)

This an SOLVE management services initial menu selection number (for example, MENU(3.1)). This is the first menu displayed after logon to SOLVE management services proper (not EASINET), provided you are authorised to display it.

If this operand is omitted, the SOLVE management services Primary Menu is displayed first, unless you are authorised for only a single function (for example, File Transmission Services). In this case, the Primary Menu is automatically suppressed and you will immediately enter that function.

Examples:

```
NMLOGON APPL(NM2 )
NMLOGON MENU( 2 )
NMLOGON AP(NM5 ) M(U) NOAUTO
NMLOGON APPL(NMT) EASINET(MAIL)
```

Return Codes:

Executing the NMLOGON command sets a return code, as follows:

- 0** No errors, session established.
- 4** Error during the initial session dialogue.
- 8** Session could not be established.

Non-zero return codes issue additional messages which detail the specific error.

Autologon

Your user ID is often the same for both TSO and SOLVE management services, and you will normally use the TSO Full-Screen Interface from the same user ID.

When an SOLVE management services user ID is defined, the TSO AUTOLOGON parameter can be specified to determine whether that user ID is authorised for automatic logon from TSO:

- ▶ If the AUTO operand is specified, or allowed to default on the NMLOGON command when the EASINET keyword is not present, SOLVE management services attempts to perform an automatic logon to bypass the normal entry of user ID and password. If the TSO user ID is defined to SOLVE management services and is authorised for TSO AUTOLOGON, the automatic logon occurs.
- ▶ If the NOAUTO operand is specified on the NMLOGON command, no automatic logon is attempted and the initial SOLVE management services logo screen is displayed for you to enter your user ID and password. This option is normally only used for those TSO users with multiple Exiting from the TSO Full-Screen Interface

Note: Where an installation uses its own security exit procedures to replace the distributed SOLVE management services UAMS functions, that exit must prescribe whether AUTOLOGON is allowed.

Operating Differences

Restrictions within TSO itself mean that you must understand some small operating differences to successfully operate in full-screen mode with the TSO Full-Screen Interface.

When you press an ENTER key or a function key, SOLVE management services displays any messages that are available for immediate presentation. Messages can arrive and be queued ready for this display. However, the TSO Full-Screen Interface cannot accumulate messages for output before some input is initiated from the keyboard.

Therefore, you are responsible for requesting any further output that may be available, by pressing either the PA1 or PA2 key. These keys have been designated to display any messages accumulated.

Note: This requirement normally only applies when operating in OCS mode.

If the number of OCS messages available for display exceeds the remaining lines on the display, SOLVE management services pauses (for the specified PROFILE command ROLL delay time) before writing more output. PA1 or PA2 must be pressed to solicit each subsequent output screen.

Unsolicited TSO messages (such as messages from other TSO users, or job end notifications) may be displayed. These may overwrite the SOLVE management services screen display.

After you have responded to the normal TSO prompt (**), SOLVE management services attempts to re-format the screen. You may have to use the PA1 or PA2 key to do this.

Whenever you suspect more output is waiting to display, or that your screen is not properly formatted, press the PA1 or PA2 key.

Note: The PA1 key functions as described above, not as with most other TSO applications. This means you cannot use PA1 to interrupt NMLOGON command processing.

The EASINET Option

The EASINET option provides access to installation-provided functions and is implemented from the EASINET NCL procedure.

This option can often avoid having to allocate SOLVE management services user IDs to users.

For example, TSO user could access the NET/MAIL feature by entering a command such as:

```
NMLOGON APPL (NMT) EASINET (MAIL)
```

assuming your EASINET procedure does recognise NMLOGON entry by using the &ZTSOUSER system variable, it then picks up the TSO user ID from the &1 user variable and the MAIL option from &2 to effect direct entry into NET/MAIL. A user does not have to have an SOLVE management services user ID.

Exiting from the TSO Full-Screen Interface

To exit from the TSO Full-Screen Interface not operating in the EASINET mode, perform a normal SOLVE management services logoff.

If NMLOGON is being used with the EASINET keyword (i.e. you are accessing the EASINET mode in your target SOLVE management services), the EASINET NCL procedure accessed will include a mechanism to terminate the session. The session can be terminated by using a CLSDST command at the appropriate processing point. For example:

```
&INTCMD CLSDST &LUNAME
```

TSO Command Interface

The TSO Command Interface provides a *line-by-line* mode of operation through the NMCMD command processor. This interface creates an environment where you can use sub-commands to route requests to the SOLVE system, which can return single line messages.

The TSO Command Interface can be executed from an on-line or batch environment by invoking the the NMCMD command from within a TSO session.

Requests are executed by NCL procedures running under the SOLVE management services *User Services* option. Any results of these requests are returned unformatted and output as single lines to your terminal (or an equivalent batch output dataset).

This interface can be used to connect to a SOLVE management services operating in a different VTAM domain. It is available in OS/MVS and MSP environments only, although it may be used to connect to SOLVE management services operating under any operating system.

Note: No disconnect or reconnect facility is available for the TSO Command Interface.

The NMCMD Command

TSO users wishing to use the NMCMD command must have access allocated to the SOLVE management services *User Services* component. The format of the NMCMD command is:

```
NMCMD    applname
          [ USER( userid [ / password ] ) ]
          [ LUPREF( luprefix ) ]
          [ LUNAME( luname ) ]
          [ INDD( ddname | NMCMDSINMD ) ]
          [ CMD( sub-command ) ]
          [ MODE | PROMPT ]
```

applname (Mandatory)

The VTAM APPL name of the SOLVE system to which you wish to connect. If this positional parameter is not present, you will be prompted to supply one.

USER(*userid*[/*password*])

Can be any valid SOLVE management services user ID (or user ID and password combination). The provision of a user ID and password is not necessary, provided the TSO user ID and SOLVE management services user ID names are identical. Also, the SOLVE management services user ID definition must allow TSO AUTOLOGON.

If a user ID is supplied without a password, NMCMD will prompt for the password in non-display mode.

LUPREF(*luprefix*)

Specify a 1–5 character prefix for the series of ACB names that NMCMD will attempt to use when connecting to SOLVE management services. A suffix of 001 is initially added to the *luname* prefix supplied, and an attempt made to open the ACB. If this ACB is in use, the suffix keeps incrementing and retrying the open until an open is successful, an unrecoverable open error occurs, or the list of defined ACBs is exhausted. If this operand is omitted, a prefix of NMTSO is used.

LUNAME(*luname*)

Specify the entire ACB name NMCMD will attempt to open to connect to SOLVE management services. If the open fails for any reason NMCMD processing terminates immediately.

INDD(NMCMDSIN | *ddname*)

Specify the *ddname* of a dataset or dataset member containing a list of NMCMD sub-commands. These are invoked using the READ sub-command (see below). Sub-commands may be continued onto the next record by specifying a '+' as the last non-blank character on a line.

CMD(*sub-command*)

May be any NMCMD sub-command. If a sub-command is supplied on the NMCMD statement, the connection to SOLVE management services is automatically terminated when sub-command processing is complete.

If no sub-command is provided on the NMCMD statement, the user enters TSO prompt mode. In this mode the session with SOLVE management services is first established before the prompt *NMREADY* appears at the terminal. A sub-command may then be entered and, once completed, the *NMREADY* prompt is received again. This mode of operation continues until the END sub-command is entered. The session then ends and the NMCMD command completes, returning control to TSO.

MODE | PROMPT

Indicates whether NMCMD is to issue its *NMREADY* message as a TSO *mode*, or *prompt* message. Some TSO facilities are sensitive to prompt output (for example, the DATA PROMPT, ENDDATA combination used within a TSO CLIST), while other facilities require mode type output such as batch TSO processing. For most TSO environments, the default MODE type of output should be suitable.

Examples:

```
NMCMDB NMPROD LUPREF(NMP) INDD(IFILE)
NMCMDB NMT LUNAME(NMTEST) USER(TUSER)
NMCMDB NM CMD(READ)
```

NMCMD Sub-Commands

The following sub-commands invoke a TSO sub-command processor. Some commands are related to the SOLVE management services FTS component and are fully documented in Chapter 8, *FTS TSO Interface*, of this manual:

FTSEND

Ad hoc request to transmit a file.

FTSDEF

Creates an FTS definition.

FTSXMIT

Transmits an existing definition.

FTSDEL

Deletes an FTS definition.

Other (non-FTS) sub-commands which may be entered are:

END

Terminates NMCMD processing, returning to TSO or the calling CLIST.

HELP

Displays standard TSO Help information for NMCMD.

MAXRC *nn*

Sets the maximum return code value, where *nn* is a number in the range 0 – 99. NMCMD terminates if any SOLVE management services procedure returns an &RETCODE value greater than the MAXRC value defined. Specifying an invalid value resets MAXRC to its default value of 16.

READ

Indicates that NMCMD should read the dataset defined by the INDD keyword (specified when NMCD was invoked) and process this as a list of sub-commands. During this process, the normal NMREADY prompt is not issued. The results returned from individual sub-commands are written to the terminal (unless otherwise suppressed through sub-command options).

MORE

To solicit any outstanding responses after a previous sub-command has been issued as an internal SOLVE management services command (see below).

any other data

Any data unrecognised as one of the above sub-commands is passed through to SOLVE management services as a command. This is discussed in more detail in the following section. (See also the 'MORE' sub-command above).

Issuing SOLVE or VTAM Commands

As mentioned above, any input not recognised as a standard sub-commands is sent to SOLVE management services.

The distributed procedure assumes it is a SOLVE (or VTAM) command and executes it by issuing an NCL &INTCMD statement specifying the command passed. The procedure uses the &INTREAD statement of NCL to retrieve the results of the command and passes them back to the NMCMD processor.

Since an unknown number of messages may be returned by the internal command, the procedure makes some assumptions as to when the last message has been received. If you suspect more messages may be available after issuing any SOLVE or VTAM commands, use the MORE sub-command of NMCMD to read any messages queued to the internal command environment in SOLVE management services.

The NMCMD Operating Environment

NMCMD executes as a TSO command processor and invokes sub-command processors for certain requests. It uses the Virtual 3270 Interface provided with the EIP feature to perform the necessary connection to the requested SOLVE management services, and all subsequent I/O activity. (See the chapter on the Virtual 3270 Interface later in this manual for details).

Communicating with the Virtual 3270 Interface through a parameter area, NMCMD passes any SOLVE management services name, and the user ID and password where these are provided. If the user ID and password are not provided, the Virtual 3270 Interface attempts a logon using the TSO user ID for the currently executing user.

After a successful logon, control is passed within SOLVE management services to the User Services option. The supplied User Services procedure \$USERSER detects the logon through this interface and executes procedure \$USERTSO to handle requests.

The \$USERTSO procedure communicates with the *Virtual 3270 Interface* using the standard full-screen panel \$USV3270 defined in SOLVE management services. This panel is effectively unformatted and consists of multiple lines of text which are eventually passed back to the

NMCMD processor. (See the chapter on the *Virtual 3270 Interface* for the additional requirements.)

If required, procedure \$USERTSO may be customised. For example, to intercept user-defined sub-commands which further enhance interface functions.

TSO Batch Execution of NMCMD

The NMCMD command processor can be used to input commands to SOLVE management services from a batch environment:

Execute TSO as a batch job (as described in the *TSO Command Language Reference* manual). Input commands are stacked in the dataset referenced by the SYSTSIN DD statement. Output is returned to the SYSTSPRT DD statement.

Return Codes:

Executing the NMCMD command sets a return code in Register 15 which can be tested by a TSO CLIST to determine the outcome of the process. Any of the following decimal return codes can be set:

- 0** NMCMD completed all processing without error.
- 4** The logon to the requested SOLVE management services failed as the specified user ID is already in use.
- 8** The logon to the selected SOLVE management services failed for some reason other than the user ID is already logged on.
- 16** NMCMD is unable to continue the dialogue through the Virtual 3270 Interface due to a logical error. An accompanying message provides further details on the precise cause.
- 20** A VTAM I/O or session error has occurred.
- 24** GETMAIN (or FREEMAIN) error.
- 32** A TSO macro has returned a failure condition.

5nn

NMCMD terminated as an SOLVE management services procedure returned with an &RETCODE value of *nn*, which is greater than the MAXRC value specified (the default MAXRC value is 16).

Non-zero return codes are always accompanied by a message explaining the reason for failure.

Examples:

The following examples show how the NMCMD TSO command processor can be used in both on-line and batch environments. Input may be entered following the usual READY prompt displayed by TSO, or from an ISPF panel which lets you enter TSO commands (for example). In these examples responses returned to the terminal are shown in **BOLD** type, and prompts are *italicised* areas.

Example 1:

The following command connects to the SOLVE management services system NMTEST, requests a file transmission, then disconnects and returns to TSO. Assuming the command was entered from within ISPF the termination results in the normal ******* sequence being displayed.

```
NMCMD  NMTEST  CMD('TRANSMIT DOCFILE')
N42506 TRANSMISSION REQUEST DOCFILE ACCEPTED.
***
```

Example 2:

In this example NMCMD is invoked from TSO and two commands are issued to send a general broadcast before ending the session.

```
READY
  NMCMD NMNY

NMREADY
  NSBRO GB1=An IPL of the production system is scheduled
    for 7pm tonight.
  N15404 LINE 1 OF GENERAL BROADCAST TEXT SET.
NMREADY
NSBRO GSEND

N15501 BROADCAST COMPLETE - MESSAGE SENT TO 231 TERMINALS.
NMREADY
  END

READY
```

Example 3:

In this example TSO is executed as a batch job and several commands input to SOLVE management services system NMSYD1.

```
//TRANS12 JOB (acct),.....  
//*  
//TSOBATCH EXEC PGM=IKJEFT01  
//SYSTSPRT DD SYSOUT=A  
//SYSTSIN DD *  
NMCMD NMSYD1 USER(BATCHUSR/PROD1) LUNAME(NMBATCH1)  
LINK START=PROD2  
XMIT ATRANS12  
XMIT SPRRUN  
LOG JOB TRANS12 COMPLETE  
END  
/*
```

Example 4:

This example shows how a list of commands can be input to NMCMD in the foreground by using the READ sub-command of NMCMD. Assume that a series of NMCMD sub-commands have been placed in member NMCOMMS in the PDS called TSO.CNTL. The entire list of commands may be executed under TSO in the following manner:

```
ALLOC DD(I) DSN(TSO.CNTL(NMCOMMS)) SHR  
NMCMD NMNY INDD(I) CMD(READ)
```

Note that these commands may be used from a TSO CLIST or an ISPF Dialogue Manager session. If required, any output messages could be suppressed. Such a procedure could check the return code from NMCMD to determine whether a failure message should be sent to the invoking user.

Batch Command Interface (BCI)

Invoke BCI by executing the NMBCI program in a batch processing environment, or by calling it from other programs.

Commands are read from an input file (in card image format) and the replies received are routed to an output file. A user *command exit* can also be called where commands may be added, changed, or deleted. Return codes from an executed NCL procedure may be translated to JCL return codes, which can then be interpreted using conditional JCL statements.

Requests are executed by NCL procedures running under the SOLVE management services User Services option, and the unformatted results of these requests are returned to the output device.

This interface may be used to connect to SOLVE management services operating within the same or within a different VTAM domain.

BCI Command Types

Commands read from an input file fall into two categories:

- ❑ BCI control commands used to provide details for session establishment, to request activation of a BCI command exit, and to disconnect the session. These commands are described in a following section.
- ❑ Any other commands, which are routed unchanged to the target SOLVE system for interpretation and execution. These are processed by NCL procedure \$USERBCI which runs under the SOLVE management services User Services option (see the following section entitled BCI Operation for details of how each request is processed).

Once a BCI command exit has been activated, all non-BCI input commands are passed for pre-processing before continuing. The exit can add new control or user commands, and can modify command contents or delete commands.

BCI Input

Commands are extracted from an input file referenced by the SYSIN DD statement and may be any LRECL=80 dataset.

Only the first 72 columns of the record can contain command data; the remainder of the record is ignored. A command may be preceded by one or more spaces and is terminated by a space. Any data (including additional blanks) after the command, is assumed to be operands or parameters associated with the command.

BCI Output

An activity log is written to an output file and consists of 121-character print lines where the first byte is a machine control character. The file is referenced by the SYSPRINT DD statement and should be specified as RECFM=FBM with LRECL=121.

The activity log records the commands processed by the BCI, the user commands (after insertion or modification by any command exit), the replies from the User Services procedure, and any messages issued by the BCI itself.

The column headed *Terminal* indicates the device that issued the command as one from the following:

- ❑ The system input file name,
- ❑ The *luname* of the virtual terminal used for the session, or
- ❑ The name of the command exit (if the command is added or modified).

An additional flag character preceding the command or message reply indicates the source as follows:

+ Indicating it was read from the system input file.

- Shows it was inserted or changed by the command exit.

= Indicates a message issued by the BCI.

A blank indicates a reply from the User Services procedure.

The BCILOGON Command

The BCILOGON command is used to establish a session with the desired SOLVE management services on behalf of the user specified.

The format of the BCILOGON command is:

```
BCILOGON applname
          USER userid ( /password | PASSWORD password )
          [ MENU menu ]
          [ LUNAME applid | LUPREF luprefix ]
```

where:

applname

Is the VTAM APPL name of the SOLVE management services to which you wish to connect. This must be the first parameter following the BCILOGON command.

USER *userid* { /*password* | PASSWORD *password* }

Provides the valid SOLVE:management services user ID and password combination which is to be used for the session. The user ID and password, separated by a slash, may be specified in a single string after the USER keyword (i.e. as USER *userid/password*), or provided separately (i.e. as USER *userid* PASSWORD *password*). Both the user ID and password must be specified if BCILOGON is run as an ad hoc batch job.

Note: If accessing BCI from another program, the existing user ID and password values for that system (as taken from the UAMS database, or the equivalent security exit procedure) are defaulted for the interface. Logging onto BCI will appear to be transparent, without you having to provide user ID and password, or the user ID of the batch job submitter.

MENU *menu*

The initial menu selection data for the SOLVE management services User Services facility. This may contain up to five characters. If present, the characters *U.menu* are passed to SOLVE management services to perform initial menu selection.

If omitted, the User Services procedure defined for the BCI user ID will be selected (this should be \$USERBCI in the BCI facility)

distributed). However, if the user is set up to use the standard \$USERSER procedure, a menu option of BCI can be specified.

LUNAME *applid* or LUPREF *luprefix*

Specify the entire ACB name (LUNAME *applid*), or a 1 to 5 character LUNAME prefix (LUPREF *luprefix*) the BCI will attempt to open to connect to SOLVE management services. If an open fails for any reason, interface processing terminates immediately.

When the LUPREF operand is used, a 3-digit number (*nnn*) is appended to the *luprefix* value. Initially *nnn* is set to 001 and an attempt to open ACB *luprefix*001 is performed. If this ACB is in use or varied inactive, the suffix keeps incrementing and the open is retried until an open is successful; or an unrecoverable open error occurs; or the list of defined ACBs is exhausted.

If neither the LUNAME nor LUPREF operand is specified, then an LUPREF of NMBCI is assumed.

Examples:

```
BCILOGON NMCDRSC USER XYZ/XYZ
BCILOGON NMMAIN USER ANET LU ABCINET PASSWORD GO
BCILOGON NMTEST USER OPER LUPREF NMTSO PASSWORD XYZZY
```

Note: Note that the BCILOGON command may be generated by a user command exit.

Other than the *applname* parameter which must be the first parameter after the BCILOGON command, subsequent parameters may be entered in any order.

The BCIDISC Command

The BCIDISC command is used to terminate the session with SOLVE management services after processing with that system is complete.

The BCIDISC command has no parameters, its format is simply:

```
BCIDISC
```

This command logs you off SOLVE management services and frees the VTAM ACB being used this session. If the end of the input file is reached and an active session exists, then a BCIDISC command is assumed.

The BCIEXITC Command

User commands may be inspected by a user *command exit* program before being subsequently processed by the BCI. The BCIEXITC command is used to activate or deactivate a command exit, and has the following format:

```
BCIEXITC [ exit-name ]
```

exit-name

Is the 1 to 8 character name of the program to be loaded as the user exit for the BCI. It must reside in a standard load library for OS system users, or within the Core-image library for VSE system users. Specifying the *exit-name* on the BCIEXITC command causes the module to be loaded only. It is not called until a command is next read from the input file.

If the exit name is omitted, any currently-executing exit is deactivated.

Only one command exit may be active at any one time. If you must invoke a different exit, first use a BCIEXITC command to deactivate the current exit.

BCI Operation

After completing a successful BCILogon command, BCI converses with a User Services procedure defined to SOLVE management services. The BCILogon command makes an initial menu selection of U (for the User Services menu).

By using the optional MENU operand of the BCILogon command, additional data can be passed to the initial procedure in order to invoke a special procedure for conversing with BCI. This initial procedure uses the &ZPSKIP NCL statement to receive and process this data in the usual manner (see your *Network Control Language User's Guide*).

As previously mentioned, BCI makes use of the Virtual 3270 Interface. The User Services procedure (and its associated panel) used for the conversation must meet certain basic requirements which are fully described in Chapter 5, *Virtual 3270 Interface*. The first screen line on the panel should contain three output fields for control information. The second line is an input command line, and the remaining lines are used to return any output data. The supplied procedure (\$USERBCI) and panel (\$USV3270) conform to these standards and should be used as a model.

Note: BCI defines itself to the Virtual 3270 Interface as a 3270 Model-4 and, while it will only send a single command on the first input line of the panel, it will accept up to 41 data lines in the response (the Model-4 type terminal contains 43 lines of 80 characters each).

After any command has been processed by the procedure, including the BCILogon command, the first two control fields are reported as the panel name and procedure name respectively. The third field is regarded as a return code, and set as &RETCODE. If this is non-zero, the user is disconnected and documentary messages written to the system output file. The job is cancelled with a JCL return code of 12.

Procedure \$USERBCI

Procedure \$USERBCI contains all the commands necessary to communicate with a virtual BCI 3270. It may also be used for a real 3270 and be extended by the installation to cater for additional functions. The procedure contains comments which adequately describe the basic functions it performs.

\$USERBCI recognises several commands (described below) as functions it must execute. Any other commands received are assumed to be SOLVE commands and are processed by specifying them on an &INTCMD

statement. This makes the procedure as powerful as the privileges associated with the particular user ID for which it is invoked, and can be restricted at the installation's discretion.

The results of an &INTCMD command are accumulated by issuing &INTREAD statements and placing the resulting messages on the output lines of the panel.

If all panel output lines are used, the data is sent to BCI by issuing an &PANEL statement. The messages are extracted and BCI awaits further output. More than one panel may be required to send the results of some commands. This process is automatically handled by the interface.

The following commands are not passed to SOLVE management services but processed by the distributed procedure \$USERBCI:

ECHO [*nn*] [*text*]

Is recognised by \$USERBCI as a test command. The *text* specified is echoed *nn* times (up to 41 maximum) in the output lines of the panel. If *nn* is omitted or non-numeric, the text is echoed 3 times. If no text is supplied, a sequence of characters is returned.

EXIT

May be used to emulate the F3 key and will end procedure \$USERBCI. This can be useful if \$USERBCI is nested from another BCINCL procedure. It is not used to terminate the BCI session - see the BCIDISC command, described previously.

INVRG [*nn*]

Is used by \$USERBCI to set &RETCODE to *nn*. If this is a non-zero value, the batch program will terminate immediately with message N96009 issued.

MORE

A request to perform another &INTREAD operation. After an &INTCMD has been issued, the results are read using &INTREAD commands. Since an unknown number of messages may be returned, the procedure makes some assumptions as to when the last message has been read. If you suspect more messages are available, these can be solicited through the MORE command. Any number of MORE commands may be issued.

NOMORE

Signifies that any outstanding messages not yet returned should be purged. This results in the procedure issuing an &INTCLEAR statement to discard any such outstanding responses.

TRACE ON/OFF

May be used to issue an NCL &CONTROL TRACE or &CONTROL NOTRACE command in the BCINCL procedure. This may be useful for debugging BCI NCL procedures.

Customising \$USERBCI

An installation may further tailor the \$USERBCI procedure in order to extend its function. For example, certain commands could be recognised to invoke other procedures or panels if desired.

If the procedure is modified by the installation, it is recommended that you test it using a real 3270 before testing it on the BCI. The procedure can be tested at any 3270 (this does not have to be a Model-4, as the number of screen lines is handled by the procedure).

Note: If a 132-column screen is used, output is truncated by BCI to the first 79 characters. When testing at a real terminal, MORE command processing can be abandoned by pressing PF12 or PF24.

For ease of testing or occasional usage, we suggest you amend the existing User Services procedure and its associated panel, to allow entry to the tailored procedure from a normal SOLVE management services session for users required to maintain the system. If so, then you won't have to log onto SOLVE management services using a BCI user ID. This eliminates a potential conflict between BCI and on-line usage.

The BCI Command Exit

When a record is read from the system input file, it is first scanned to determine whether it is a BCI control command. If not, it is seen as being a user command to be passed to SOLVE management services. However, before it is passed it is reformatted to remove leading blanks. If a command exit is active, it is called passing the command as a parameter. The command exit may then choose to modify the command, to reject it, or to insert additional commands into the input stream.

On entering the exit, register 1 contains the address of a 76-byte area as follows:

Bytes 1-4

Contain a binary full word indicating the length of the following data which can be a maximum of 72 bytes.

Bytes 5-76

Contain the user command and data with leading blanks removed and any unused characters to the right filled with blanks.

Before returning, the exit should set Register 15 to indicate what action BCI is to take with the returned data, as follows:

Return Codes:

0

The command is to be passed to SOLVE management services. The actual command may have been modified and its length changed by updating the 76-byte area as passed as the parameter to the exit. The reformatted command is reported on the system print file (except for the BCILOGON command as noted below).

4

The command is to be passed to SOLVE management services (as for R15=0 above). However, after it has been fully processed, the exit is invoked again instead of reading the next card from the system input file. Therefore, this return code is used to insert commands. The user command area presented on the subsequent invocation contains the command as returned on the last such call, and not the original command.

8

The command is to be ignored, and is not reported.

The command exit can be used to generate a BCILOGON command to automatically generate user ID and password data. If a BCILOGON command is returned, only the command and SOLVE management services *applname* are reported on the system print file.

The command exit is especially useful for removing superfluous commands that may be generated by a library system (for example). However, any user commands before the BCI command is detected are ignored, so the BCIEXITC command itself may exist in a library member and be detected correctly.

As the command exit is loaded by the BCIEXITC command and merely executed on each subsequent command until deactivated, the program is effectively resident. Therefore, data areas may be defined within the exit to control calls to the exit during insertion processing. Additionally, if a short command is returned in the command area, the rest of the area may be used by the exit since it is neither inspected, amended, or reported by BCI. If the length field is greater than 72, it is assumed to contain 72. Similarly, if the length is zero or negative, it is assumed that no command is present and no action is taken. However, if the return code is 4, the exit is invoked again. In both instances, the length field is not modified by BCI.

An example command exit is supplied - see Appendix B, *Installation Considerations* for further details.

JCL Return Codes

When the BCI program terminates, it sets a JCL return code which may be tested for successful completion. The return code will have one of the following values:

0 No errors detected.

4 BCI has processed all input cards and issued warning messages. Check the messages to ensure that BCI has made the correct assumptions.

8 BCI has processed all input cards and has found that serious errors and incorrect results are likely. Check the messages and correct the input.

12 BCI has detected serious errors processing all input cards. The results are incomplete. Check the message and correct the input.

1001-1009

The BCI NCL procedure has ended with a non-zero return code (&RETCODE). The JCL return code is the value in &RETCODE plus 1000. For example, if &RETCODE is set to 2, then the JCL return code returned will be 1002. BCI will issue message N96009 and terminate the job with the JCL return code.

U4095

BCI has detected a serious problem. The BCI output file will contain messages that indicate the nature of the error; it may be an internal error. A dump is produced which may be required for problem diagnosis by your product supplier.

Example:

This example shows the same job as Example 1, but is set up for execution on an OS operating system.

```
//NMBCI      JOB      etc
//NMBCI      EXEC    PGM=NMBCI
//SYSPRINT   DD     SYSOUT=A
//SYSIN      DD     *
BCILOGON NMTEST USER TESTBCI/TESTPSWD MENU BCI LU NMDOSBCI
TRANSMIT TESTFILE
SHOW USERS
LIST $USERBCI
EXEC TESTPROC
BCIDISC
BCILOGON NMACCTS PASSWORD MONEY USER WAGES
TRANSMIT PAYROLL
BCIDISC
/*
//
```

Virtual 3270 Interface

The Virtual 3270 Interface allows an external program to communicate with SOLVE management services functions, primarily NCL procedures, using full-screen dialogue.

Requests are normally passed to NCL procedures running under the SOLVE management services User Services option. The interface user program and NCL procedure can determine the state and flow of data on the conversation as they desire.

This interface may be used to connect to SOLVE management services operating in the same or a different VTAM domain.

Using the Virtual 3270 Interface

Various calls to the interface are supported to simplify session establishment and termination, and for reading and writing the information contained in 3270 datastreams.

A parameter list supplied as the DSECT \$NMVTAP in the SOLVE management services distribution library is used for communication. Once output has been received on the session a *Field Descriptor List* is built and the address placed in the main parameter area. This allows the interface user program to examine the contents of the panel and to update input fields for return to SOLVE management services.

The NMV3270I load module can be linked to a program requiring interface services, or loaded as part of an initialisation process. Calls to the interface are then supported by passing the initialised parameter area to provide the following functions:

Open

Starts a session with SOLVE management services using parameter list information to determine the target system *applid*, the local VTAM ACB name to be used for the session, the user ID and password for the user requesting the logon, the virtual 3270 screen size, and any initial menu selection required.

If an LUNAME prefix is used for the ACB name (APPLID) and the ACB fails to open because the APPL has been varied inactive, the operation retries using the next generic ACB name. This process is repeated until an open is successful, or until five successive inactive or unknown ACB name errors occur.

Receive

Is used to receive a full-screen of data and builds a Field Descriptor List which describes all input (and optionally output) fields found on the panel. This may then be examined by the calling application to determine the contents of the various fields.

Send

Is used to return data to SOLVE management services. The Field Descriptor List may be plugged with the address of input data which is to be placed in panel input fields. A code specifying the function key to be pressed can also be set (the default is the ENTER key).

Send, then Receive

Performs both send and receive functions before returning control to the calling program. This is useful when issuing a request which expects to solicit some data in reply.

Send, then Terminate

performs a Send then awaits session termination before returning control (for example, after a LOGOFF command has been sent). If the session does not shut down in an orderly fashion, then it is disconnected before control is returned to the calling application.

Terminate

Closes the session immediately without further I/O and returns to the caller.

More documentation may be found in the DSECT \$NMVTAP describing the fields which can be set for the various calls, and those which may be set by the interface on return.

The supplied DSECT \$NMFLDDS can be used to map the data returned in the Field Descriptor List obtained from the panels used in a conversation. See those DSECTS in the distributed NMMACLIB, and the procedure documentation in this manual, for precise details on how to use this facility.

Controlling the Conversation

By default, NMV3270I selects the SOLVE management services User Services menu as the initial menu selection. This can be modified on the Open call. However, no matter what function gets control and issues the first I/O from SOLVE management services, it is the responsibility of the program calling the interface and the NCL procedure (or other function) at the SOLVE management services end to agree to some protocol and maintain proper synchronisation.

The Virtual 3270 Interface provides services which simplify VTAM session communication, and let you analyse the data being passed across the session using a 3270 full-screen conversation.

Interfacing to NCL Procedures

Once a session has been established through the Virtual 3270 Interface and has made an initial menu selection that passes control to an NCL procedure, the procedure begins the conversation by issuing an &PANEL statement. This results in a 3270 datastream being sent to the interface.

The interface user must issue a *Receive* call at which time the virtual panel image and Field Descriptor List are built and returned to the caller.

By requesting a *Send* (or a *Send then Receive*) function, any input is returned to SOLVE management services with the appropriate attention key set. This satisfies the &PANEL statement at which time the procedure can examine the input fields through their token names, and the key through the &INKEY system variable in the usual manner.

Panel Characteristics

While any panel definition could be used for this conversation, the simpler the panel definition the easier it is for the interface user to process. Panel \$USV3270 is used by both the Batch Command Interface and the TSO Command Interface and uses specific conventions. The

panel is defined as a Model-4 (43 line) panel, the first line containing three protected fields as follows:

- ▶ The first field contains the characters \$USV3270 and can be used to verify that the interface user is connected to an NCL procedure designed to handle this type of communication.
- ▶ The second field contains the system variable &0. On output this will be substituted and contains the name of the currently executing procedure. This helps maintain synchronisation throughout the conversation.
- ▶ The third field contains the system variable &RETCODE which can be used to determine the status of processing within the procedure.

The contents of these fields are extracted and placed into defined areas of the parameter list passed between the user program and the interface, regardless of whether the user requested all or only input fields be built in the Field Descriptor List. Hence there is an advantage in maintaining this first line convention for any application.

The second line of panel \$USV3270 is an input field and contains the token name &REQUEST. This of course appears to the user program as the first input field in the Field Descriptor List and hence is easily accessible. Normally this would be used to describe the type of function that the interface user is requesting from the NCL procedure.

Subsequent lines of the panel are all input lines and may be used to contain qualifying input from the user application, or for returned text data from the NCL procedure.

Where any external application is required to use the services of the Virtual 3270 Interface, we recommend that the procedures and panels described above be used as models to assist development.

FTS External Interfaces Overview

The File Transmission Services (FTS) feature of SOLVE management services provides many sophisticated functions for the movement of datasets between CPUs. It is intended primarily for those installations that require the movement of datasets on a frequent and consistent basis, where possibly the same datasets are transmitted every night.

However, some installations may need a simpler mechanism for transmitting datasets on an ad hoc basis, and when customising panels presented to particular users.

The FTS NCL Interface

The FTS NCL Interface is a suite of NCL procedures that extend the scope of FTS. These procedures can cater for ad hoc requests and can be tailored, along with the supplied full-screen panels, to suit your requirements.

Such procedures present an alternative for requesting FTS transmissions for users who are signed on to SOLVE management services. Menus and full-screen selection lists provide user convenience, and user access can be restricted as necessary. However, these procedures do not offer the full function provided by the standard FTS panels (accessed through option 3 of the SOLVE : Primary Menu and are not intended as a replacement.

Support for TSO Users

Installations which require further support for TSO users include an alternative for requesting FTS functions in the NMCMD command processor. This is invoked from TSO and supports sub-commands for the purpose of requesting FTS services.

FTS requests originating from NMCMD processing under TSO are forwarded and processed by the FTS NCL Interface procedures mentioned above. This lets TSO users request file transmission activity without connecting to SOLVE management services directly.

In addition, a set of CLISTs, panel definitions, and programs designed to operate under ISPF's Dialogue Manager, extend the function for TSO users. These also use NMCMD for routing requests to SOLVE management services and include additional NCL procedures and SOLVE management services utilities for performing services according to the FTS request type.

The following chapters contain more details on the FTS facilities mentioned above: Further details on the NMCMD command processor are available in Chapter 3, *TSO Command Interface*, of this manual, while the various requirements for implementation are discussed in Appendix B, *Installation Considerations*.

NCL Interface Operational Environment

FTS NCL Interface procedures and panels are designed to be used with FTS V3.0, in conjunction with SOLVE V3.0.

All FTS facilities available from TSO require the installation and tailoring of the EIP V3.0.

FTS Procedures

Transmitting a dataset using the base FTS feature is a two-step process:

- ▶ Firstly, *defining* the transmission. This involves completing up to three full-screen panels via the FTS definition maintenance menu. Transmission Definitions are stored on the SOLVE management services VFS VSAM database.
- ▶ The second step is the actual file Transmission Request: The definition provided in the first step is *requested* via the XMIT (or TRANSMIT) command from OCS or the Transmission Supervision menus. When this request is entered, FTS takes a copy of the definition, places it on the VFS database, and creates another record to maintain checkpoint information, etc. The transmission then proceeds when possible (once an FTS initiator is available, the INMC link is active, etc.).

Transmission Status Information

The status of file transmissions can be obtained from a number of sources:

- ▶ The SHOW FTS command from OCS shows active file transmissions and the status of links to other systems.
- ▶ FTS Transmission Supervision menus include options to display active file transmissions and to list requests held on the VFS database, including details of failed transmissions, completed transmissions, queued transmissions, etc.

The SHOW FTS command can be executed via &INTCMD from an NCL procedure and the results used to produce a tailored fullscreen panel. The other interfaces, however, are in a fixed format.

NCL Access to VFS

NCL procedures have access to VSAM datasets (termed user databases, or UDBs) via &FILE statements. A UDB can be in either internal format or external format. External format UDBs are those usually created by a means other than NCL procedures, but NCL procedures can read and update these datasets.

To access a UDB, an NCL procedure issues an &FILE GET statement specifying a logical file name that has been previously assigned using a UDBCTL command.

This mechanism does not normally allow an NCL procedure to access an SOLVE management services dataset. However, SYSPARMS SYSUDB operand allows an NCL procedure to access an SOLVE management services VSAM file, such as VFS. The SYSUDB operand is entered as follows:

```
SYSPARMS SYSUDB=YES
```

An NCL procedure may then issue the statement:

```
&FILE GET VFS EXT-HEX
```

to access to the VFS dataset. In this mode of operation, which is reserved for accessing system datasets, no UDBCTL command is required. The name on the &FILE GET statement is the DD name (OS/VS) or file name on the DLBL card (VSE). The NCL procedure concerned must then access VFS in *external mode*, as specified by the EXT-HEX operand shown in the example above.

Having gained access to VFS, NCL procedures can read and write FTS-related records and therefore supplement the FTS base system. The additional facilities this provides includes writing transmission definitions, retrieving transmission status information, providing tailored displays. etc. However, you will need to have a detailed knowledge of FTS record formats and internals.

WARNING: Extreme care must be taken if the supplied procedures are modified. Invalid records written to VFS could severely impact the execution of FTS, and cause an abend.
Records other than those manipulated by the supplied procedures should not be modified.

Details of NCL access to UDBs in *external format* are documented in your *Network Control Language User's Guide*.

FTS NCL Interface

This chapter details the functions performed by the distributed procedures, and the various calling sequences. Documentation within each procedure details the various parameters required on entering a procedure, and the information available on exit.

These procedures are used from within SOLVE management services to provide an alternate means of requesting FTS transmissions.

Some procedures described in this chapter are also used by the TSO FTS Interface and the ISPF Dialogue Components. These components use additional NCL procedures to perform various functions and the relevant chapters later in this manual contain details on the actual procedures used.

The &FTS Variables

A number of parameters must be supplied to transmit a dataset successfully. For example, when a supplied procedure requires these parameters to write a transmission definition record to VFS, the parameters must be supplied in a number of local variables whose names start with the characters &FTS. When a procedure retrieves a definition from VFS, the options are made available in the &FTS variables.

&FTS variables are listed next, together with edit rules that must be applied to each one. The supplied procedures ensure these rules are followed. All variables correspond to the options contained on the

standard transmission definition panels. The supplied procedures set each one to the required length before being written to VFS:

Note: If new procedures are written, or the supplied procedures modified, it is the installation's responsibility to ensure that all edit rules are followed. Failure to do so causes unpredictable results.

&FTSNAME

Must be the 1 to 12 character transmission definition name.

&FTSDEST

Must be the name of the destination SOLVE management services, 1 to 12 characters long.

&FTSFRDSN

The name of the dataset to be transmitted, which should conform to standard dataset naming conventions. Dynamic allocation of the dataset fails if this is not the case.

&FTSTODSN

The name of the receiving dataset, which should conform to standard dataset naming conventions. Dynamic allocation of the dataset fails if this is not the case.

&FTSINDD

The DD name (OS/VS) or file name of the DLBL (VSE) card in the execution JCL for the dataset to be transmitted, if dynamic allocation is not to be used. Either &FTSFRDSN or &FTSINDD should be supplied, but not both.

&FTSOUTDD

Is the DD name (OS/VS) or file name of the DLBL card (VSE) in the execution JCL for the receiving dataset, if dynamic allocation is not to be used. Either &FTSTODSN or &FTSOUTDD should be supplied, but not both.

&FTSSTAT

Must contain either S to indicate a system definition, or P to indicate a private definition.

&FTSCLASS

Must contain the transmission class as a letter (from A to Z).

&FTSTRACK

Must contain either Y to indicate the user must acknowledge successful dataset transmission, or N to indicate acknowledgement is not required.

&FTSFRCMD

Can contain a command (up to 40 characters) to be executed at the sending end when the transmission completes. Note that commands can only be executed for system definitions.

&FTSTOCMD

Can contain a command (up to 40 characters) to be executed at the receiving end when the transmission completes. Note that commands can only be executed for system definitions.

&FTSRCACK

Must contain either Y to indicate the user must acknowledge successful dataset receipt, or N to indicate acknowledgement is not required.

&FTSLUSER

Contains the *user ID* of the user who last executed a transmission definition. Not required when defining or modifying a definition, but is made available when retrieving a definition.

&FTSLDATE

Contains the date when the definition was last executed. Not required when defining or modifying a definition, but is made available when retrieving a definition.

&FTSLTIME

Contains the time when the definition was last executed. Not required when defining or modifying a definition, but is made available when retrieving a definition.

&FTSAUTOR

Must contain Y if the transmission can be automatically restarted after link failure, N if not.

&FTSEMIN

Must contain Y if the input dataset is to be emptied after dataset transmission, or N.

&FTSFRVOL

Can contain the name of the volume where the dataset to be transmitted resides if not cataloged. If &FTSFRVOL is supplied, &FTSFRUNT must also be supplied.

&FTSFRUNT

Can contain the unit name of the volume on which the dataset to be transmitted resides if not cataloged. If &FTSFRUNT is supplied, &FTSFRVOL must also be supplied.

&FTSFRDSP

Must contain the disposition to be used when the dataset is transmitted, as either OLD or SHR.

&FTSTOCKO

Must contain Y if the output dataset must be empty before the transmission can proceed, or N.

&FTSTOVOL

Can contain the name of the volume on which the output dataset is to be allocated if it is to be allocated with a disposition of NEW or REP. Either &FTSTOVOL or &FTSTOUNT, or both, must be specified for a NEW (or REP) dataset. Otherwise, no volume, unit, or space information should be supplied.

&FTSTOUNT

Can contain the name of the unit on which the output dataset is to be allocated if it is to be allocated with a disposition of NEW or REP. Either &FTSTOVOL or &FTSTOUNT, or both, must be specified for a NEW (or REP) dataset. Otherwise no volume, unit or space information should be supplied.

&FTSTOSPC

Can contain the space allocation type for a NEW or REP output dataset as either CYLS, TRKS, or BLKS. If &FTSTOSPC is supplied, &FTSTOPRI, &FTSTOSEC, and &FTSTODIR must also be supplied. None of these variables should be supplied if the dataset is not allocated with a disposition of NEW or REP.

&FTSTOPRI

Can contain the primary space allocation for a NEW or REP output dataset as a 1 to 5 digit number. If &FTSTOPRI is supplied, &FTSTOSPC, &FTSTOSEC, and &FTSTODIR must also be supplied. None of these variables should be supplied if the dataset is not allocated with a disposition of NEW or REP.

&FTSTOSEC

Can contain the secondary space allocation for a NEW or REP output dataset as a 1 to 5 digit number. If &FTSTOSEC is supplied, &FTSTOSPC, &FTSTOPRI, and &FTSTODIR must also be supplied. None of these variables should be supplied if the dataset is not allocated with a disposition of NEW or REP.

&FTSTODIR

Can contain the number of directory blocks for a NEW or REP output dataset as a 1 to 5 digit number. If the output dataset is to be a sequential dataset, &FTSTODIR must contain zero. If &FTSTODIR is supplied, &FTSTOSPC, &FTSTOPRI, and &FTSTOSEC must also be supplied. None of these variables should be supplied if the dataset is not allocated with a disposition of NEW or REP.

&FTSTORFM

Can contain the RECFM of the output dataset if it is allocated as NEW or REP, or where an image-copy transmission is not required. The RECFM value may be F, FB, FBA, FBM, FA, FM, V, VB, VBA, VBM, VA, VM, VS, VBS or U. It is possible for the supplied procedures to support other variations of RECFM if required (within the scope of the standard FTS). Contact your supplier. If &FTSTORFM is supplied, &FTSTOLRL and &FTSTOBLK must also be provided.

&FTSTOLRL

Can contain the LRECL of the output dataset if it is allocated as NEW or REP, or where an image-copy transmission is not required. The LRECL must be a 1 to 5 digit number. If &FTSTOLRL is supplied, &FTSTORFM and &FTSTOBLK must also be provided.

&FTSTOBLK

Can contain the BLKSIZE of the output dataset if it is allocated as NEW or REP, or where an image-copy transmission is not required. The BLKSIZE must be a 1 to 5 digit number. If &FTSTOBLK is supplied, &FTSTORFM and &FTSTOLRL must also be provided.

&FTSTODSP

must contain the disposition of the output dataset as SHR, OLD, MOD, NEW or REP. REP is treated as NEW except that if the dataset exists already it is deleted and replaced. Volume and space information is required for datasets allocated as NEW or REP.

&FTSTIMCO

Must contain Y if an image-copy transmission is required, or N.

&FTSSOCLS

Must contain the SYSOUT class if &FTSTODSN is specified as SYSOUT.

&FTSSOCOP

Must contain the number of SYSOUT copies required as a 1 to 3 digit number if &FTSTODSN is specified as SYSOUT.

&FTSSOHL D

Must contain Y if SYSOUT output is to be placed in HOLD, or N if &FTSTODSN is specified as SYSOUT.

&FTSSODST

Can contain a SYSOUT destination name as 1 to 8 characters if &FTSTODSN is specified as SYSOUT.

&FTSSOFLD

Must contain Y if SYSOUT is to have the fold option, or N if &FTSTODSN is specified as SYSOUT.

&FTSSOFRM

Can contain a 1 to 4 character form name if &FTSTODSN is specified as SYSOUT.

&FTSSOFCB

Can contain a 1 to 4 character FCB name if &FTSTODSN is specified as SYSOUT.

&FTSSOUCS

Can contain a 1 to 4 character UCS name if &FTSTODSN is specified as SYSOUT.

&FTSSOPGM

Can contain a 1 to 8 character program name to process SYSOUT if &FTSTODSN is specified as SYSOUT.

&FTSGROUP

Can contain the 1 to 8 character group identifier to be propagated with the file transmission request. This field is passed to the user dataset access authorisation exit (DSNCHECK) and is also written to the log when the transmission terminates.

&FTSUDATA

Can contain up to 40 characters of user data to be propagated with the file transmission request. This field is passed to the user dataset access authorisation exit (DSNCHECK) and is also written to the log when the transmission terminates.

&FTSFCXMT

Can contain the 1 to 12 character name of an existing FTS definition which is to be chained for transmission after the successful completion of the request being defined. By using this field serially, dependant chains of transmissions may be defined and initiated by requesting transmission of the first in the chain. As one transmission completes successfully, this field is checked and if a name is present the internal equivalent of a TRANSMIT command is issued. Private definitions may also make use of this facility as normal FTS privilege checking applies.

&FTSDELDF

Can contain Y, C, or N to indicate whether this FTS definition is to be deleted after a transmission of this name completes. When set to Y, the definition is deleted when the request completes, even if transmission failed. If set to C, the definition is to be conditionally deleted (that is, deleted only if the transmission was successful). The default setting of N means the definition is not deleted. If FTS determines the definition has been modified since requesting the transmission, then no deletion takes place regardless of the setting of this field.

NCL Procedure Descriptions

A number of NCL procedures are supplied to form the standard *FTS NCL Interface*. A description of each procedure follows, together with details of calls made to other procedures.

\$FTSMENU

Displays a menu with three options:

1. Transmit a File (calls \$FTSDEMO).
2. File Transmission Status (calls \$FTSSTAT).
3. File Transmission Definition MAINTENANCE (calls \$FTSMNT).

\$FTSDEMO

Provides an example of the type of tailored display now available, and the way in to handle ad hoc requests. The procedure requests that you enter only a name for the transmission (which must start with the your user ID), the destination, the transmission class, and the input and output file names and dispositions. This information is then edited, and defaults taken for all other options. The procedure then:

- ▶ Calls \$FTSDPUT to write a transmission definition record to VFS as a private definition specifying conditional definition deletion.
- ▶ Executes an XMIT command to request transmission, and waits for a response.
- ▶ Returns the response to the XMIT command to your panel.

SFTSSTAT

Provides a full screen display of FTS status information. The REFRESH field on the panel lets you nominate a refresh period of 1 to 99 seconds. When the period expires, the display is automatically refreshed with current information. A refresh rate of 0 freezes the display. The display is also refreshed if the ENTER key is pressed.

The information displayed is obtained from a number of sources: Request records are read from VFS, the status of active transmissions is obtained from an &INTCMD SHOW FTS command, and the status of links to other SOLVE management services is obtained from an &INTCMD SHOW LINKS command. If the display is longer than the displayed window, it can be scrolled up and down using standard F7 or F8 keys. The information field can be scrolled left or right using the standard F10 or F11 keys. The information is displayed in a number of columns, with headings:

NAME

The transmission name.

DESTN/ORGN

The name of the destination or originating SOLVE management services. If the link to this SOLVE management services is not available, the name is shown in high intensity (monochrome displays), or will blink red (seven-colour display).

D

Transmission direction: O means outbound, I means inbound.

C

Transmission class.

STATUS

The current status of the transmission.

INFORMATION

Varies according to transmission status. Can include failure reasons, completion statistics, or (for active transmissions) the current character and block counts.

To obtain more information about a particular request, place an S can be placed alongside the name of the request, and press ENTER. SFTSDSST is called to provide another display with dataset names and other information. You can only display those requests whose names start with your user ID.

Delete a transmission request from the database by entering D alongside the request and pressing ENTER or a scroll key. If the request name starts with your user ID and is not currently active, it is deleted by calling \$FTSDLST.

\$FTSMNT

Provides an alternative to the use of the base FTS transmission definition maintenance. This procedure can be used in place of the base FTS support to provide definition selection list capabilities. Select or delete definitions using a scrollable selection list.

All panels displayed are tailorable, and so can be shortened, or certain defaults used, etc.

Options are available to modify, add, or delete transmission definitions. The supplied procedures will render any new or modified definition a private definition. \$FTSMNT calls \$FTSSEL to provide selection list capabilities and definition modification, \$FTSADD to add a new definition, and \$FTSDDEL to delete a definition.

\$FTSADD

Sets defaults for certain options, then prompts the user with the standard definition panels before writing a new definition to VFS. Calls \$FTSDSFR to display a From panel, \$FTSDSTO to display a To panel, \$FTSDSSO to display a SYSOUT panel, and \$FTSDPUT to write a definition record from the &FTS variables built from your input.

\$FTSDDEL

Deletes a transmission definition record from VFS.

\$FTSDGET

Retrieves a transmission definition record from VFS and makes the options available in the &FTS variables.

FTSDLST

Deletes a transmission request from VFS, unless it is currently active.

\$FTSDPUT

Writes a transmission definition record to VFS, using the options available in the &FTS variables.

\$FTSDSFR

Prompts you with standard From panel and primes the &FTS variables with the information obtained from it.

\$FTSDSPL

Retrieves a transmission definition record from VFS by calling \$FTSDGET, then displays the standard definition panels by calling \$FTSDSFR, \$FTSDSTO and \$FTSDSSO. Modifications made by the user are placed in the &FTS variables and \$FTSDPUT is called to write the modified definition record to VFS.

\$FTSDSSO

Prompts user with standard SYSOUT panel and primes the &FTS variables with the information obtained from it.

\$FTSDSST

Retrieves a transmission request from VFS and displays a panel containing the dataset names involved in the request.

\$FTSDSTO

Prompts user with standard To panel and primes the &FTS variables with the information obtained from it.

\$FTSSEL

Retrieves all transmission definition records from VFS and displays a scrollable selection list. You may select a definition for modification (in which case \$FTSDSPL is called), or can delete a definition (in which case \$FTSDDEL is called). In addition, if the passed definition name already exists, \$FTSDSPL is called immediately for modification of the definition.

Limitations of FTS Procedures

The following details some of the limitations that apply to the implementation of the FTS procedures:

- ▶ Dataset passwords are not supported.
- ▶ The editing implemented within the NCL procedures is not as comprehensive as that currently provided by base FTS. Depending on the functions utilised, this can lead to errors at transmission time that currently may be picked up at definition time. For example;

Where dataset names are not edited comprehensively, which could lead to dynamic allocation failures.

To dataset DCB information is not edited comprehensively – there is no cross-check between LRECL, BLKSIZE and RECFM to ensure consistency. This could lead to dataset OPEN failures.

No cross-checks are made between Partitioned Dataset allocation and the provision of a member name (with the exception of the ISPF Dialogue Components). This could lead to dataset OPEN failures.

- ▶ Only specific RECFM combinations are supported. Other combinations may be possible. If required, contact your supplier.

Customising NCL Procedures

This section provides some indications of the types of tailoring that may be applied to the supplied NCL procedures, and how they can be used:

Name Masking

\$FTSSTAT and \$FTSDEMO procedures implement name masking, so that a user can only request a transmission, or display, or delete a transmission request whose name begins with their user ID.

If you wish to eliminate this masking technique, remove the `&MASK=&USERID` statement at the beginning of those procedures, or set the mask to an alternative value from 1 to 12 characters using a standard assignment statement.

Status Display Filtering

The \$FTSSTAT procedure displays the status of FTS as a whole. It is a simple matter to filter this display so that only those transmissions whose names start with the user's user ID (for example) are displayed. As explained above, this procedure already only lets a user select or delete a request whose name starts with their user ID.

Dynamic Transmission Definition Generation

The \$FTSDEMO procedure dynamically generates a transmission definition, based partly on user's input and partly on prescribed defaults. The new definition is added specifying conditional definition deletion, and a TRANSMIT command issued to request the transmission. Provided the transmission completes successfully, the definition is deleted automatically. Otherwise it remains and may be amended and requested again in the usual manner. Alternatively, procedure \$FTSDEMO may be modified so that definition deletion occurs regardless of the transmission's success (by setting `&FTSDELDF` to Y). In this case, the same steps used to originally define the request may be required to recover from failed transmissions.

These techniques can be utilised in a number of ways. The information necessary to create a transmission definition can be obtained from a variety of sources, such as a UDB, from full-screen panels, or from parameters passed to an NCL procedure. For example, an NCL procedure based on \$FTSDemo could be passed only the names of the input and output datasets to set defaults for all other options before requesting the transmission.

Such a procedure can also be invoked from a batch job via UTIL0001. The step of the batch job that executes UTIL0001 could submit the command:

```
F NM,EXEC FTSSUB INPUT.DATASET OUTPUT.DATASET
```

which has the effect of executing the NCL procedure FTSSUB under control of the SYSOPER user ID, passing it the names of the input and output datasets. This procedure then defines a transmission and requests it.

Private and System Definitions

As previously discussed, the supplied procedures always write transmission definitions as *private definitions*. This is accomplished by setting the &FTSSTAT variable to P in the following \$FTSADD, \$FTSDemo, and \$FTSDSPL procedures.

These procedures can be modified to set a value of S in the \$FTSSTAT variable, if required.

FTS TSO Interface

The NMCMD Command Processor

TSO users may use the supplied NMCMD TSO command processor to enter FTS requests directly from TSO. The NMCMD command processor provides a general interface between TSO and the NCL procedures executing under the SOLVE management services User Services option.

Having invoked the NMCMD processor, various sub-commands are available. Some of these are of a general nature and are described in Chapter 3, *TSO Command Interface*, of this manual. Others are specific FTS related sub-commands and are documented in this chapter. All sub-command functions are initially edited by TSO sub-command processors before being forwarded to SOLVE management services where the standard FTS NCL Interface procedures are called to process the requests.

The FTS Sub-Command Environment

As described in the *TSO Command Interface* chapter, the NMCMD command processor can be invoked during your TSO session to perform on-line requests, or from a batch job by executing TSO in the background. The primary function of the NMCMD processor is to contact the specified SOLVE management services by logging on through the TSO Interface. This logon occurs in background mode and

you need not be aware of the session between TSO and SOLVE management services. Once this session is established any of the NMCMD sub-commands may be entered for processing by SOLVE management services.

After entering an FTS sub-command, the FTS sub-command processor of NMCMD is called to edit the request. It ensures that all required operands are present and performs cross-editing functions. The request is then forwarded to the NCL procedure \$USTSFTS running under User Services. This procedure creates the &FTS variables required for each request and calls the appropriate FTS NCL procedure (as described in the chapter on *FTS NCL Interface*) to carry out the function.

Messages returned by the processing NCL procedures provide the initial status of each FTS request. These are written to your TSO terminal by issuing TSO PUTLINE macros, but may be suppressed if desired.

Note: All FTS functions invoked by NMCMD sub-commands execute the standard FTS NCL Interface procedures. Hence any restrictions imposed by these procedures (other than those imposed by FTS itself) will also apply to these TSO sub-commands.

If a transmission fails for some reason, then you will normally have to clear the condition and restart the failed request from within the FTS Supervision option. Alternatively, other means may be provided by an installation (for example, through additional NMCMD commands to delete requests).

The NMCMD FTSDEF Sub-Command

The FTSDEF command lets TSO users create and store FTS definitions in the SOLVE management services VFS database. No file transmission activity results from the definition, but a subsequent transmit request can be used to request the transmission (see the FTSXMIT sub-command).

```
FTSDEF  NAME( ftsname )
        [ COPY( ftsdef ) ]
        DEST( ftsdest )
        [ CLASS( ftsclass ) ]
        FRDSN( fromdsn )
          [ FRDSP( OLD | SHR ) ]
          [ FRVOL( fromvol ) ]
          FRUNT( fromunit ) ]
        [ MEMBERS( mem1 [ /ren1 ], mem2 [ /ren2 ], ... ) ]
        TODSN( todsn )
          [ TODSP( OLD | SHR | MOD | NEW | REP ) ]
          [ TOVOL( tovol ) ]
          [ TOUNT( tount ) ]
          [ TOSPC( units pri sec dir ) ]
          [ TORFM( toecfm ) ]
          TOBLK( tblksize ) ]
          TOLRL( tolrecl ) ]
        [ TIMCO( Y | N ) ]
        TODSN( 'SYSOUT' )
          [ SOCLS( class ) ]
          [ SOCOP( copies ) ]
          [ SOHLD( N | Y ) ]
          [ SODST( dest ) ]
          [ SOFLD( N | Y ) ]
          [ SOFRM( forms ) ]
          [ SOFCB( fcb ) ]
          [ SOUCS( ucs ) ]
          [ SOPGM( program ) ]
        [ MSG | NOMSG ]
        [ MSG | NOMSG ]
        [ PRIVATE | SYSTEM ]
        [ FRCMD( 'from command' ) ]
        [ TOCMD( 'to command' ) ]
        [ GROUP( groupid ) ]
        [ UDATA( 'user data' ) ]
        [ FRXMIT( ftsdef ) ]
```

NAME(*ftsname*) (Mandatory)

Provides the name to be assigned to this File Definition. The name can be up to 12 characters long. However, if the MEMBERS keyword (see description below) is also specified, then the specified name should not exceed 10 characters.

This name must be unique within the SOLVE management services to which you are connected otherwise the request will fail. As distributed, the FTS procedures restrict all such names to only those beginning with the user's *user ID*, regardless of the user's FTS privileges. Check with your installation representative or Systems Administrator for any FTS name restriction changes.

COPY(*ftsname*)

Can provide the name of an existing FTS transmission definition to copy. If used, the only other operands which need to be specified are those differing from the copied transmission definition. However, DEST(*), TODSN(*), and FRDSN(*) must be specified to copy the *destination*, *to dataset*, and *from dataset* respectively since those operands are mandatory. Note that if using the COPY operand, it should be preceded only by the NAME operand - any intervening operand values are lost.

DEST(*ftsdest*) (Mandatory)

Provides the 1 to 12 character name of the destination SOLVE management services where this FTS transmission is to be sent.

CLASS(*ftsclass*)

Can provide the FTS transmission class for this request (any character A to Z). If not specified, then the default is class A, unless the MEMBERS operand is also specified, when the default is class M.

FRDSN(*fromdsn*)

The name of the 'From' or source dataset to be transmitted to the target system. This can be any valid dataset name and may include a PDS member name, unless the MEMBERS keyword is also specified (see the MEMBERS operand below for details). When using TSO dataset prefixing, the first level qualifier may be omitted as it will be automatically inserted by the command processor, *unless* the dataset name is enclosed in quotation marks. If the dataset is not catalogued, FRVOL and FRUNT operands must be present.

FRDSP(OLD | SHR)

The disposition of the *fromdsn* dataset.

FRVOL(*fromvol*)

Specifies the volume-serial information of the *fromdsn* dataset if it is not catalogued. If this operand is specified, the FRUNT operand must also be present.

FRUNT(*fromunit*)

Specifies the unit information (for example, TAPE, SYSDA, etc) of the *fromdsn* dataset if it is not catalogued. If this operand is specified then the FRVOL operand must also be present.

MEMBERS(*mem1* [/*ren1*], *mem2* [/*ren2*], ...)

Allows a member list to be entered for PDS datasets. This operand is mutually exclusive with the specification of a member in the *fromdsn* dataset name. Up to 99 members may be present in the list. Any members can be renamed in the target dataset by specifying the list entry as *mem/ren* where *mem* is the member name in the *fromdsn* and *ren* is the new name to be given in the *todsn*. Each member in the list will be transmitted in a separate FTS transmission. The *ftsname* provided by the NAME operand should not exceed 10 characters if the MEMBERS operand is used, since each actual request after the first one will generate a name consisting of the *ftsname* suffixed by two digits (02–99).

TODSN(*todsn*), or TODSN('SYSOUT') (Mandatory)

Receive the transmitted file. If the name specified is 'SYSOUT' then the *fromdsn* dataset will be transmitted directly to the JES spool. In this case no other operands which begin with TO should be present, but those beginning with SO may provide any required SYSOUT options. If a *todsn* dataset name is supplied it must be compatible with the *frdsn* according to the usual FTS requirements.

TODSP(OLD | SHR | MOD | NEW | REP)

The disposition of the To, or target dataset. Unless NEW or REP is specified then the *todsn* entered must be a catalogued dataset on the destination SOLVE management services. Otherwise the TOVOL, TOUNT, TOSPC, TOLRL, TOBLK, and TORFM operands must provide the new dataset information. The REP option is treated as NEW except that the dataset will be *deleted* and *replaced* should it exist already.

TOVOL(*tovol*)

Provides the volume-serial information of the *todsn* when TODSP is NEW or REP. The new dataset will be created and catalogued on this volume. If this operand is omitted for a new dataset then the TOUNT operand must be specified and the destination, SOLVE management services, will create the dataset on any suitably authorised volume for the *tounit*.

TOUNT(*tounit*)

Provides the unit information of the *todsn* when TODSP is NEW or REP. If not supplied for a new dataset then the TOVOL operand must be present and the default unit value for the *tovol* is used.

TOSPC(*units pri sec dir*)

Provides the units of space allocation and the various quantities for a new dataset only (that is, where TODSP is specified as NEW or REP). The units may be one of CYLS, TRKS, or BLKS to request cylinder, track, or block allocation respectively.

The quantities *pri*, *sec*, and *dir* may be any 1–5 digit numbers which provide the primary, secondary, and directory block allocation quantities. Both units and *pri* must be present, while *sec* and *dir* default to zero. Unless a PDS member transmission was requested, no directory block information should be supplied.

TOLRL(*tolrecl*)

Provides the logical record length as a 1 to 5 digit number for new *todsn* datasets or where an image copy is not requested (that is, TIMCO=N is specified). If TOLRL is specified then TOBLK and TORFM must also be present.

TOBLK(*tblksize*)

Provides the block size as a 1 to 5 digit number for new *todsn* datasets or where an image copy is not requested (that is, TIMCO=N is specified). If TOBLK is specified, then TOLRL and TORFM must also be present.

TORFM(*torecfm*)

Provides the record format where the *todsn* is new or an image copy is not requested (that is, TIMCO=N is specified). It may be any one of F, FB, FBA, FA, FM, V, VB, VBA, VBM, VA, VM, VS, VBS or U. If TORFM is specified, then TOLRL and TOBLK must also be present.

TIMCO(Y | N)

Indicates whether an image-copy transmission is requested. If Y is specified then the DCB attributes of the *todsn* will be copied from the *fromdsn*, otherwise they must be supplied. The default for image-copy is Y except where the DCB information is supplied (in *tolrl*, *tblksize* and *torecfm*) in which case it is set to N.

SOCLS(*class*)

Can provide the SYSOUT class if TODSN(SYSOUT) is specified, and defaults to A.

SOCOP(*copies*)

Can provide the 1 to 3 digit number of SYSOUT copies required if TODSN(SYSOUT) is specified, and defaults to 1.

SOHLD(N | Y)

Indicates whether the SYSOUT dataset is to be placed in HOLD when TODSN(SYSOUT) is specified.

SODST(*dest*)

Can provide the 1 to 8 character SYSOUT destination if TODSN(SYSOUT) is specified.

SOFLD(N | Y)

Indicates whether the output is to have the FOLD option when TODSN(SYSOUT) is specified.

SOFRM(*forms*)

Can provide the 1 to 4 character SYSOUT forms name if TODSN(SYSOUT) is specified.

SOFEB(*fc*)

Can provide the 1 to 4 character SYSOUT FCB name if TODSN(SYSOUT) is specified.

SOUCS(*ucs*)

Can provide the 1 to 4 character SYSOUT UCS name if TODSN(SYSOUT) is specified.

SOPGM(*program*)

Can provide the 1 to 8 character program name to process SYSOUT if TODSN(SYSOUT) is specified.

MSG | NOMSG

Indicates whether messages returned from SOLVE management services should be written to the TSO terminal using PUTLINE macros. Error messages concerning session I/O cannot be suppressed by this operand.

PRIVATE | SYSTEM

Determines whether this transmission request is to be considered a PRIVATE or SYSTEM request. The default is PRIVATE. Note that the user's FTS privilege does not affect the definition of this transmission, but the actual TRANSMIT (or XMIT) command invokes user ID transmission request checking.

FRCMD('*from command*')

Specifies the *from command* or command to be executed at the transmitting end after *successful* completion of a *system definition* file transmission. Up to 40 characters may be entered (within quotation marks if embedded blanks are used). Any SOLVE command valid for entry from this source may be entered, including the EXEC and XMIT commands. Normal command authority checking applies.

TOCMD('*to command*')

Specifies the *to command* or command to be executed at the receiving end after *successful* completion of a *system definition* file transmission. Up to 40 characters may be entered (within quotation marks if embedded blanks are used). Any SOLVE command valid for entry from this source may be entered, including the EXEC and XMIT commands. Normal command authority checking applies.

GROUP(*groupid*)

Provides the 1 to 8 character user group identifier that this request will carry. This information is propagated with the file transmission request for the benefit of the FTS dataset access authorisation exit (DSNCHECK) and is written to the log at the completion (successful or otherwise) of the transmission.

UDATA('*user data*')

Specifies up to 40 characters (within quotation marks if embedded blanks are present) of user data which is associated with the request. This information is propagated with the file transmission request to the FTS dataset access authorisation exit (DSNCHECK) and is written to the log at the completion (successful or otherwise) of the transmission.

FRXMIT(*ftsdef*)

Specifies the name of an existing FTS definition which is to be chained for transmission from the current one after its successful completion. Normal FTS privilege checking applies.

The NMCMD FTSDDEL Sub-command

The FTSDDEL command lets TSO users delete existing FTS definitions from the SOLVE management services VFS database.

FTSDDEL	NAME(<i>ftsname</i>)
	[<u>NOCHAIN</u> CHAIN]
	[<u>MSG</u> NOMSG]

NAME(*ftsname*) (Mandatory)

Provides the name of the FTS definition which is to be deleted. Note that the FTS procedures only allow deletion of definition names which begin with the requesting user's user ID, regardless of their FTS privileges.

NOCHAIN | CHAIN

By specifying CHAIN the requested definition plus all definitions chained from it are deleted. For example, if an FTSDEF request is used to define a multi-member transmission definition chain, then all such definitions can be deleted by requesting an FTSDEF specifying the original definition name and the CHAIN keyword.

MSG | NOMSG

Indicates whether messages returned from SOLVE management services should be written to the TSO terminal using PUTLINE macros. Error messages concerning session I/O cannot be suppressed by specifying this operand.

The NMCMD FTSEND Sub-command

The FTSEND command provides TSO users with an ad hoc file transmission capability:

```
FTSEND NAME( ftsname )
        [ COPY( ftsdef ) ]
        DEST( ftsdest )
        [ CLASS( ftsclass ) ]
        FRDSN( fromdsn )
            [ FRDSP( OLD | SHR ) ]
            [ FRVOL( fromvol ) ]
            FRUNT( fromunit ) ]
        [ MEMBERS( mem1 [ /ren1 ], mem2 [ /ren2 ], ... ) ]
        TODSN( todsn )
            [ TODSP( OLD | SHR | MOD | NEW | REP ) ]
            [ TOVOL( tovol ) ]
            [ TOUNT( tounit ) ]
            [ TOSPC( units pri sec dir ) ]
            [ TORFM( torecfm ) ]
            TOBLK( tblksize ) ]
            TOLRL( tolrecl ) ]
        [ TIMCO( Y | N ) ]
        TODSN( 'SYSOUT' )
            [ SOCLS( class ) ]
            [ SOCOP( copies ) ]
            [ SOHLD( N | Y ) ]
            [ SODST( dest ) ]
            [ SOFLD( N | Y ) ]
            [ SOFRM( forms ) ]
            [ SOFCB( fcbl ) ]
            [ SOUCS( ucs ) ]
            [ SOPGM( program ) ]
        [ MSG | NOMSG ]
        [ MSG | NOMSG ]
        [ PRIVATE | SYSTEM ]
        [ FRCMD( 'from command' ) ]
        [ TOCMD( 'to command' ) ]
        [ GROUP( groupid ) ]
        [ UDATA( 'user data' ) ]
        [ FRXMIT( ftsdef ) ]
```

The FTSEND command services ad-hoc file transmission requests by performing the following in sequence:

FTSDEF

To create a File Transmission definition (or chained series of definitions) in the connected SOLVE management services system.

FTSXMIT

Requesting transmission of the previously-defined request (or request chain).

These functions are performed by invoking the relevant NCL procedures described in the chapter on *FTS NCL Interface*. Therefore any restrictions imposed by these procedures also apply to FTS requests originating from TSO.

The operands for the FTSEND sub-command are identical to those for the FTSDEF sub-command already described. See the FTSDEF sub-command operand descriptions for details.

The NMCMD FTSXMIT Sub-command

The FTSXMIT command lets TSO users request that an existing FTS definition be transmitted. The definition may have been added through the FTSDEF sub-command, or through the normal FTS options provided by SOLVE management services.

```
FTSXMIT NAME( ftsname )  
          [ MSG | NOMSG ]
```

where:

NAME(*ftsname*) (Mandatory)

Provides the name of the FTS definition which is to be transmitted.

MSG | NOMSG

Indicates whether messages returned from SOLVE management services should be written to the TSO terminal using PUTLINE macros. Error messages concerning session I/O cannot be suppressed by specifying this operand.

TSO Interface NCL Procedures

\$USERSER

Is the standard User Services procedure distributed with the system. It detects a logon from the External Interface through NMCMD and passes control to \$USERTSO. See the chapter on *TSO Command Interface* for more details on this facility.

\$USERTSO

Handles the first level of communication between the NMCMD TSO Command Interface processor and the requested NCL-based functions.

\$USTSFTS

Is called from \$USERTSO when an FTS request is received. For each of the FTS sub-commands of NMCMD, described earlier in this chapter. The primary purpose of the \$USTSFTS NCL procedure is to extract the data received from the TSO command processors and then call the appropriate FTS NCL Interface procedure to perform the actual function requested. By default, procedure \$USTSFTS specifies conditional definition deletion for all FTS definitions added. Processing depends on the actual FTS request, as follows:

FTSDEF

- ▶ All &FTS variables are primed with the values defined to the FTS sub-command of NMCMD, or defaults taken.
- ▶ Procedure \$FTSDPUT is called to add a new definition to the VFS database. By default, the definition must be new and will be rejected with an appropriate message if a definition with this name already exists.
- ▶ The results of the request are returned in a message to the user.
- ▶ Where a member list is provided, the above steps are repeated for each member. The initial request uses the name supplied in the &FTSNAME variable and subsequent requests are named uniquely by qualifying this name with a 2-digit suffix (02 to 99), as required. Each definition is then chained to the next using the &FTSFCXMT field. The user values supplied for the variables &FTSFCXMT, &FTSFCXMT, and &FTSFCXMT are only placed in the last definition in the chain.

FTSXMIT

- ▶ Only the variable &FTSNAME is required.
- ▶ An &INTCMD XMIT command is issued for the name provided and the returned message checked for successful completion.
- ▶ The results of the request are returned in a message to the user.

FTSDEL

- ▶ Only the variable &FTSNAME is required.
- ▶ The procedure \$FTSDDEL is called to delete the definition named. If the CHAIN keyword was also provided, \$FTSDDEL reads each record first to find the next in the chain before deleting the current record.
- ▶ A message is returned to the user indicating the success of the request, and for chained definition deletion the number of definitions deleted.

FTSEND

- ▶ Performs steps as though an FTSDEF request were immediately followed by an FTSXMIT request. If multiple members are chained together, an FTSXMIT is performed only for the first definition. A message is returned to the user indicating the status of the request.

Messages Returned to TSO

The distributed procedures return messages to the TSO environment following the execution of each FTS sub-command. If the sub-command used specified the NOMSG option then these messages are suppressed and will not be written to the terminal.

Note: Some messages returned are SOLVE messages which are received by the procedure following the execution of an internal command. Others are generated by the FTS procedures and are described below.

PFTS01 VFS *access* FAILED, KEY=*filekey* , FDBK=*fb*

This message is returned when a severe file error occurs and provides the access as READ or UPDATE, the filekey causing the error and the VSAM feedback information *fb*. The procedure terminates with a return code of 24.

PFTS02 ONLY FIRST 99 MEMBERS PROCESSED, OTHERS IGNORED.

This is a warning message only indicating that the member list received specified more than 99 members. Only the first 99 members are processed, the rest are ignored.

PFTS03 NO MEMBERS FOUND WHEN MEMBER LIST SPECIFIED.

Indicates a request to process a member list was received but the member list was empty. The procedure terminates with return code 24.

PFTS04 TRANSMISSION REQUEST *name* IN ERROR.

This general message is issued when an error is encountered and will be followed by another message providing the error details.

PFTS05 PRIVATE DEFINITIONS MAY NOT SPECIFY FRCMD OR TOCMD.

The acceptance of a FRCMD or TOCMD is restricted to system definitions only. If set for a private definition, the definition is rejected and the procedure terminates with return code 8.

PFTS06 DEFINITION REQUESTED TO COPY WAS NOT FOUND.

A request specified an existing definition to copy but no definition by that name was found. The request is rejected and the procedure terminates with return code of 20.

PFTS07 PROC=*procname*, REQUEST=*request*, INVALID FOR MEMBER LIST.

The request received is not supported when a member list is also specified. This indicates an invalid call to procedure *procname* which terminates with return code 24.

PFTS08 INVALID REQUEST '*request*' RECEIVED BY *procname*.

The request indicated is not supported by the procedure named.
Processing terminates with a return code of 24.

PFTS09 FTS REQUEST *name* DEFINED. *nn* REQUESTS CHAINED FOR TRANSMISSION.

This is an informational message which follows the successful processing of a member list and provides the total number of definitions successfully defined.

PFTS10 DEFINITION *name* ADDED.

This is an informational message which follows the successful addition of a single FTS definition.

PFTS11 DEFINITION *name* ALREADY EXISTS, IS NOT ADDED.

An attempt was made to define the definition indicated but one by that name already existed. The request is rejected and the procedure terminates with return code 16.

PFTS12 DEFINITION *name* DELETED.

This is an information message returned following the successful deletion of a single FTS definition.

PFTS13 DEFINITION *name* NOT DELETED, RECORD NOT FOUND.

A request to delete the definition named failed as no definition by that name was found. The request fails and the procedure terminates with return code 12.

PFTS14 *nn* CHAINED DEFINITIONS DELETED.

A request to delete all definitions chained from the definition name provided has completed. This message is an information message providing the number of successful deletions. If the chained deletion was terminated by an error, another message will be issued indicating the error condition.

Return Codes From \$USTSFTS

Various return codes are set by procedure \$USTSFTS depending upon the outcome of the request. These are returned to TSO and can be used to terminate an NMCMD session if the MAXRC value of NMCMD is exceeded. In this case the procedure return code (a number in the range 0 to 99) is prefixed by a 5 (producing a number in the range 500 to 599) which becomes the actual return code of NMCMD on termination.

More details on the MAXRC sub-command of NMCMD, and the use of return codes from NMCMD may be found in the chapter on *TSO Command Interface* in this manual. The return codes produced by \$USTSFTS are described below. All non-zero return codes are accompanied by one or more messages which are described in the previous section:

- 0** Request was processed successfully.
- 8** Attempt to supply a FRCMD or TOCMD (command for execution in the From or To system after a transmission completes successfully) for a *private* definition. The definition is rejected as this facility is restricted to *system* definitions.
- 12** Indicates an attempt to delete a definition which does not exist. The request is rejected.
- 14** An FTSXMIT request was issued but was not successful. The reason for the failure is found in an accompanying message.
- 16** An attempt to create a new definition was rejected as a definition by that name already exists.
- 20** An FTSDEF or FTSSEND request specified the name of a definition to copy but this definition did not exist. The request is rejected.
- 24** A VFS file error or some other severe logic error has occurred. Processing cannot continue.

Examples of FTS Requests Using NMCMD

The following examples show how the NMCMD TSO command processor and the FTS sub-commands may be used when requesting FTS services. Input may be entered following the usual READY prompt displayed by TSO, or perhaps from an ISPF panel allowing entry of TSO

commands. Responses returned to the terminal are shown in **BOLD** type, prompts in *ITALICISED* areas.

Example 1

The following command will connect to the SOLVE system NMTEST, request an ad hoc file transmission to system NMPROD, then disconnect and return to TSO. Assuming the command was entered from within ISPF the termination displays the normal ******* :

```
NMCMD  NMTEST  CMD( 'FTSSSEND
      NAME(U03ISMEMO) DEST(NMPROD)
      FRDSN(MEMO.TEXT(M324))
      TODSN(U99NET.MEMO.TEXT(M324)) ' )
```

```
N42506 TRANSMISSION REQUEST U03ISMEMO ACCEPTED.
***
```

Note that you need not specify the input dataset prefix if profiled from TSO to use a dataset prefix. Only the required operands are shown in this example, all others assuming the default values.

Example 2

In this example NMCMD is invoked from TSO and two commands issued before ending the session. The first command uses the **MEMBERS** operand to transmit three members from a PDS and rename two in the target dataset. The second command copies an existing definition but directs the output to the dataset specified:

```
READY
NMCMD NMNY
NMREADY
FTSSSEND  NAME(X55) DEST(NMSANFRAN)
      FRDSN(USER.DATA)
      MEMBERS(TRANS1/BATCHX,TRANS2/BATCHY,CTLCARD)
      TODSN(PROD.BATCH.CARDS)
      FRDSP(SHR) UDATA(BAT123)
PFTS09 FTS REQUEST X55 DEFINED. 3 REQUESTS CHAINED FOR
TRANSMISSION. N42505 REQUEST FOR X55 QUEUED - LINK NOT
ACTIVE
NMREADY
FTSSSEND  NAME(XA55PGM) COPY(XA55SFPGM)
      DEST(*) FRDSN(*)
      TODSN(XA55.TEST.LOAD)
      TOCMD('EXEC J55')
N42505 REQUEST FOR XA55PGM QUEUED - LINK NOT ACTIVE
NMREADY
END

READY
```

Example 3

In this example TSO is executed as a batch job and several commands input to the SOLVE system NMSYD1.

```
//NMCMD      JOB      (acct),.....
//*
//TSOBATCH EXEC  PGM=IKJEFT01
//SYSTSPRT DD    SYSOUT=A
//SYSTSIN  DD      *
NMCMD  NMSYD1  USER(BATCHUSR/PROD1)
FTSXMIT NAME(ATRANS12)
FTSXMIT NAME(SPRRUN)
FTSDEF NAME(NEWBATCH) DEST(NMSYD2) +
FRDSN(PROD.BATCHRUN) +
TODSN(PROD.COPY.DATA) +
CLASS(P) TODSP(NEW) TOVOL(PROD04) +
TOSPC(CYLS 5 2) TIMCO(Y) +
FRXMIT(REPORT1)
FTSDEF NAME(REPORT1) DEST(NMSYD2) +
FRDSN(PROD.BATCHRUN.LIST) +
TODSN('SYSOUT') SOCLS(H) SOHLD(Y) +
FRCMD('MSG OPER BATCH 1 TRANSMISSION COMPLETE')
FTSXMIT NAME(NEWBATCH)
END
//
```

Example 4

This example shows how a list of commands can be input to NMCMD in the foreground by using the READ sub-command of NMCMD.

Assume the following NMCMD sub-commands have been placed in member NMCOMMS in the PDS called TSO.CNTL.

```
FTSDEF  NAME(USER01X) DEST(NMLA) +
        FRDSN(TSO.DATA) TODSN(LA.STAGEIN) +
        UDATA('USER01 SYSA') NOMSG
FTSXMIT NAME(USER01X) NOMSG
FTSXMIT NAME(USER01RUN) NOMSG
```

This series of commands may then be invoked in the following manner:

```
ALLOC DD(I) DSN(TSO.CNTL(NMCOMMS)) SHR
NMCMD NMNY INDD(I) CMD(READ)
```

Note that these commands can be used from a TSO CLIST and since all the FTS sub-commands specified the NOMSG option, no output to the terminal results. The procedure could check the return code from NMCMD before sending a message to the invoking user.

FTS ISPF Dialogue Interface

Various EIP constitute an ISPF Dialogue application providing FTS services. This chapter describes each of these components, how they interact within the dialogue application and the steps required to tailor the system to suit the particular needs and standards of an installation.

Primarily the dialogue is used to gather information about a requested file transfer before invoking NMCMD to connect to the SOLVE management services that will perform the transmission. A full understanding of these mechanisms requires that you be familiar with *FTS NCL Interface* procedures, and the *FTS TSO Interface* capabilities described in previous chapters of this manual. For information concerning the actual installation requirements of the dialogue see the appendix on *Installation Considerations* in this manual.

Note: In this chapter, it is assumed that the FTS feature has already been installed as part of your SOLVE system, unless it is stated otherwise.

Functional Overview

TSO and ISPF play a major role in systems development and maintenance. Quite often these functions require the movement of files from one system to another. The FTS ISPF Dialogue Interface is designed to integrate into the installation's existing ISPF environment and thus provide a convenient means of effecting such file transmissions.

By using the TSO facilities available, extensive editing is performed before the request is forwarded to SOLVE management services. Although the dialogue contains some additional functions not available when using the standard FTS feature within SOLVE management services, it does not implement all FTS options and is not intended as a replacement of the standard FTS facility.

Any SOLVE management services in the network that is accessible from the TSO host can be used to perform file transfers. The choice of system depends on the location of the source dataset. If the source dataset resides on the TSO host system, then the transfer is said to be from the *local* host and will be effected by SOLVE management services running on the same CPU. Where the source dataset is on some other host system then it is referred to as a *remote* host and the transfer is effected by SOLVE management services running on that remote CPU. In either case, the target or receiving system may be any other SOLVE management services in the network which also has the FTS feature.

Although it is possible for the dialogue to request file transfers from any system in the network, certain functions are restricted to local host transfers and often to specific dataset types. These are outlined below. A more detailed description of all dialogue functions is provided later in this chapter.

Direct File Transfers

Datasets with formats that can be directly read by the FTS feature of SOLVE can be used for *direct* file transfer. This means that the source dataset is accessed at the time of the transmission (which may itself be some time after it was originally requested), and the contents transferred directly into an existing or newly-created dataset in the target system.

This type of transmission is restricted to PDS members and sequential files (of various record formats) only.

Staged File Transfers

In a *staged* file transfer the source and target datasets are not accessed for the transmission itself. Rather the contents of the source dataset are moved into a temporary *staging dataset*, usually converting the dataset format in the process. This occurs when the request is issued while the user waits a few moments. The actual file transmission then proceeds from this staging dataset to another staging dataset at the receiving end. When the transmission is complete, a utility is attached to move contents into the actual target dataset and the temporary staging datasets are deleted.

This technique allows dataset types not normally supported for direct file transmission, to be copied from one system to another. The dialogue application provides such support for transmitting PDS load library members, entire PDS datasets and VSAM datasets.

The use of staged file transfers is restricted to datasets from the local system and to existing target datasets only.

Edit Restrictions

When requesting a direct file transmission from a remote system it is not possible for the dialogue application to fully check all details concerning the source dataset, or in fact to even check whether the dataset exists. In addition it may make certain assumptions as to the nature of the dataset format, depending upon the request and context, etc. (as described more fully, later in this chapter).

Dialogue Components

The various ISPF Dialogue Manager procedures, Assembler programs, panels, Help panels, and message members, plus supporting NCL procedures.

Dialogue Procedures

NMFTS

Is the main entry procedure which controls the dialogue flow. It calls procedure NMFTSID which sets installation options then presents the FROM and TO panels to collect information regarding the user's file transmission request. When all such information is available, it invokes one of three other procedures (NMFTSC, NMFTSM or NMFTSS, depending on the transfer option requested) to perform any actions related to the type of transfer.

NMFTSC

Is invoked when the C option (direct copy) is selected and all information regarding systems and datasets is available. If the source dataset is a PDS on the local system and no member name was supplied, then a full member selection list is supplied. The TSO command processor NMCMD is then invoked to schedule the transmission request with the appropriate SOLVE management services.

NMFTSM

Is invoked when the M option (direct copy specifying multiple members) is selected, and all information regarding systems and datasets is available. The source dataset must be a PDS (verified if on the local system) and a panel is presented to let you enter up to 20 member names and new names, where they are to be renamed. The TSO command processor NMCMD is then invoked to schedule the transmission request with the appropriate SOLVE management services.

NMFTSS

Is invoked when the S option (staged copy) is selected and all information regarding systems and datasets is available. If the source dataset is a PDS on the local system and no member name was supplied, then a full member selection list is supplied. For VSAM files a panel allowing selected key ranges (applicable to KSDS datasets) is presented. After allocating the necessary staging datasets, the TSO command processor NMCMD is invoked to schedule the transmission request with the local SOLVE management services.

NMFTSID

Is an initialisation procedure called from NMFTS when first invoked. It allows the installation to specify various tailoring options and to define the systems available for FTS functions. See the section on *Tailoring the Dialogue* later in this chapter for a further description.

NMFTSED

Is called from NMFTS to edit the system names provided by the user for the FTS functions. It verifies that the system is valid and that the target system is accessible from the source system, based on the information supplied in procedure NMFTSID.

Dialogue Programs

NMFTDSRG

Is an Assembler program which is called to determine the dataset organisation of a requested dataset. It expects the dialogue variable &DSN1 to be in the shared variable pool and to contain the name (qualified or unqualified) of the dataset in question. It sets a return code as follows:

0 Function successful.

8 Dataset not catalogued.

For a return code of zero, NMFTDSRG sets the following dialogue variables in the shared variable pool:

&ZDSORG

The dataset organisation as one of the following:

PS – physical sequential,
PO – partitioned organisation,
VS – VSAM organisation,
IS – indexed sequential (not supported)
DA – direct access (not supported)

&VOL1

Contains the actual volume on which the dataset resides.

&ZRECFM

Contains the record format of the dataset.

&ZLRECL

Contains the logical record length for the dataset.

&ZBLKSZ

Contains the dataset block size.

&ZUNITS

Contains the dataset allocation units as CYLINDERS, TRACKS, or BLOCKS.

&ZPRISPC

Contains the primary space allocation quantity.

&ZSECSPC

Contains the secondary space allocation quantity.

NMFTMSEL

Is a dialogue program which uses ISPF Dialogue Manager Table handling to provide a full member selection list for a PDS dataset. It then processes selected members. Depending on the call parameters, it either updates a sequential file with IEBCOPY control statements, or constructs the dialogue variable &MEMSEL to reflect the selections (in the format expected by the MEMBERS operand of the FTSEND sub-command of NMCMD). Members may be selected for file transfer with an S to copy the member, or an R to copy with rename, in which

case a new member name must be supplied. The following parameters may be passed to NMFTMSEL.

DD1=

Provides the DD name of the source dataset for the file transmission. This indicates the dataset from which the member selection list will be built. If not present, a DD name of SYSUT1 will be assumed.

DD2=*ddname2*

Provides the DD name of the target dataset for an IEBCOPY unload operation (normally the staging dataset). If not specified, a DD name of SYSUT2 is assumed.

CTL=*ddname3*

Provides the DD name of a dataset which can be opened for update to contain the IEBCOPY control statements generated following user selections.

If not supplied, no dataset is assumed. The dialogue variable &MEMSEL will contain the selections in the format expected by the MEMBERS operand of the FTSEND sub-command of NMCMD.

MEM=*member*

May contain a single member name (for a single member unload operation) or an asterisk (*), which indicates that all members are selected. In either case, if a blank *member* value is entered then a member selection list is presented.

Dialogue Panels

NMFTFR

Is the FROM panel which prompts for source dataset and system details displayed on entering the system by procedure NMFTS.

NMFTTO

Is the TO panel which requests target dataset and system details and is displayed by procedure NMFTS.

NMFTNEW

Is displayed by procedure NMFTS when the target dataset is specified to have a disposition of NEW.

NMFTMSEL

Is used by the program NMFTMSEL to display the member selection list.

NMFTSMM

Is used by procedure NMFTSM to allow the entry of multiple member names when option M is selected for this request.

NMFTVSE

Is a confirmation panel. When a VSAM source file is requested to be transmitted to a new target file an IDCAMS *temporary export* operation is assumed. This panel allows confirmation of this request before proceeding.

NMFTVSR

Is displayed by NMFTSC when a source VSAM dataset is being transmitted to an existing target VSAM dataset to allow the entry of key ranges restricting the records selected.

NMFTDSTG

Is displayed by procedure NMFTSC when allocation of a new staging dataset fails due to the fact that it already exists. Options are provided to continue with the dialogue and retry the operation, delete the dataset and retry, or leave the dataset and try allocating a different one.

NMFTWAIT

Is displayed at your terminal when some delay may be expected while the transmission is scheduled.

NMFTSERR

Is displayed when the transmission request is aborted due to error. Displays any messages returned.

NMFTSOK

Is displayed when the transmission was scheduled successfully.

NMFTHxxx

Help panels.

Dialogue Messages

NMFTM00, NMFTM01 and NMFTM02 contain ISPF messages used by the dialogue.

NCL Procedures

The following NCL procedures are used by the ISPF Dialogue and each is invoked either from the FRCMD or TOCMD operand on the NMCMD command. This means they will be executed on successful completion of the requested FTS transmission in the From or sending end, and the To or receiving end, respectively.

\$@COPY

Is invoked from either \$@L or \$@M (see below) after successfully completing the staged transmission of an unloaded PDS dataset. This procedure is executed to reload the transmitted members into the actual target dataset and calls UTIL0009 to attach IEBCOPY.

\$@DELETE

Is invoked via FRCMD to delete a staging dataset in the transmitting system and notify the TSO user of the successful transmission completion.

@E

Is invoked via TOCMD to import a VSAM dataset from a staging file which was transmitted as a temporary exported dataset.

\$@L

Is invoked via TOCMD to reload a PDS from a staging dataset. This procedure is used when member replacement was not requested and merely sets a parameter before calling procedure \$@COPY to perform the operation.

\$@M

Is invoked via TOCMD to reload a PDS from a staging dataset. This procedure is used when member replacement was requested and merely sets a parameter before calling procedure \$@COPY to perform the operation.

\$@NMFTS

A tailoring procedure which is called from several other NCL procedures to set any installation dataset naming options.

\$@NOTIFY

Is invoked via FRCMD to notify the TSO user of the successful completion of the FTS transmission.

\$@R

Is invoked via TOCMD to reload a VSAM file following a staged transmission through the standard IDCAMS REPRO function. This procedure allows record replacement and merely sets a parameter before calling procedure **\$@REPRO** to perform the operation.

\$@REPRO

Is invoked from either **\$@R** or **\$@S** and calls UTIL0007 to attach the IDCAMS system utility and perform a REPRO load operation.

\$@S

Is invoked via TOCMD to reload a VSAM file following a staged transmission through the standard IDCAMS REPRO function. This procedure does not allow record replacement and merely sets a parameter before calling procedure **\$@REPRO** to perform the operation.

SOLVE Management Services Utilities

UTIL0007

A utility program called from NCL which attaches the system utility IDCAMS and passes control statements from NCL as parameters (see the documentation within the utility source for further details).

UTIL0009

A utility program called from NCL which attaches the system utility IEBCOPY and may override the default DDnames through parameters supplied from NCL (see the documentation within the utility source for further details).

Tailoring the Dialogue

After installing the above components into the correct execution libraries (see the appendix on *Installation Considerations* for further details about this process) you must perform some tailoring of the dialogue. The ISPF Dialogue procedure NMFTSID and the NCL procedure \$@NMFTS are specifically available for this purpose.

Procedure NMFTSID

This procedure is executed when the dialogue is first entered and is used to set various installation options as well as describing the operational environment of the system in which it is running. Parameters set by this initialisation procedure are placed into the shared variable pool where they may be accessed by other components in the dialogue. The following variables may be set by the installation:

SYSID

Must contain the name by which the local system (on which this procedure will run) is to be known. For example, if the dialogue is to be used on two systems, commonly known as PROD and TEST, then a copy of this procedure must be available on each system with &SYSID set to PROD and TEST as required.

LCLPREF

Contains the dataset prefix to be used for any temporary files, such as staging datasets and IEBCOPY sysin control files, required by the application on this local system. By default this is set to the TSO user's standard TSO prefix but may be changed to any suitable prefix and can contain more than one qualifying name (for example, SYSFTS.&SYSUSER). Since certain unrecoverable transmission failures can result in temporary datasets not being deleted, it may be useful to provide an initial prefix in this manner which clearly identifies this type of dataset for later clean-up processing. Further considerations for this parameter are discussed in the section on *Managing Temporary Datasets* below.

LCLUNIT

Can contain the name of a generic unit on which any local temporary datasets are to be placed. It is set to SYSDA by default.

LCLVOL

Can contain the name of a specific volume on which any local temporary datasets are to be placed.

REMPREF

Contains the dataset prefix to be used for any temporary files, such as staging datasets, required by the application on a remote system. By default this is set to the TSO user's standard TSO prefix but may be changed to any suitable prefix and can contain more than one qualifying name (for example, SYSFTS.&SYSUSER). Since certain unrecoverable transmission failures can result in temporary datasets not being deleted, it may be useful to provide an initial prefix in this manner which clearly identifies this type of dataset for later clean-up processing. Further considerations for this parameter are discussed in the section on *Managing Temporary Datasets* below.

REMUNIT

Can contain the name of a generic unit on which any remote temporary datasets are to be placed. It is set to SYSDA by default.

&REMVOL

Can contain the name of a specific volume on which any remote temporary datasets are to be placed.

VSAMSPC

Can contain the cylinder allocations for staging datasets if VSAM staged file transmission are to be used. Specify both the primary and secondary extents as shown in the procedure where the default of (5 5) is set.

FTSTYP

Nominates the type of FTS transmission that is to be used by this application and can be either SYSTEM or PRIVATE. The default is SYSTEM which means that users must have FTS SYSTEM Request privilege in SOLVE management services. Because the FTS definitions are created by NCL procedures, users need not have FTS SYSTEM Definition privilege. If the installation requires that PRIVATE requests only be used then the NCL procedure \$USTSFTS must be modified (see the appendix on *Installation Considerations*).

MSGOPT

Is used to determine the type of message handling required by the dialogue. For TSOE installations it is recommended that you use a default value of E, which means that the dialogue will trap error messages where possible and display them on full-screen panels. A value of Y means that messages will be written to the terminal followed by *** in the usual ISPF manner. A value of N suppresses message delivery.

&SYSFID n , &SYSTID n , &NMAPPL n and &NMLINK n

Are used to inform the dialogue of the valid system combinations available to the user for file transfer. Each combination, identified by n (which must be serially allocated from 1 onwards), requires all four variables and represents an SOLVE management services from which a file transmission may be requested. The variable &SYSFID n is the name by which a *from* or source system is known to the user (for example, PROD or TEST) and is nominated by the user when specifying the FROM system for the file transfer.

Where the SYSFID n used in this manner has the same value as variable &SYSID (described above) then the file transfer is deemed to be from a local system and staged transmissions will be permitted. The variable &SYSTID n describes a *to* or target system which is accessible from &SYSFID n . Note that a given *from* system may have more than one target system, in which case an additional set of variables is required to describe each such combination. The variable &NMAPPL n provides the network name of SOLVE management services running on the *from* system, which can be contacted to perform the file transmission, while &NMLINK n provides the name of the link from that SOLVE management services that can be used to transmit the dataset to the target (i.e. &SYSTID n) system. Code as many sets of these variables as are required for your configuration, starting with $n=1$.

Reverse (or bi-directional) links require two such definitions to allow transmission to be sourced from either end. See the procedure itself for examples of these parameters.

Procedure \$@NMFTS

This NCL procedure is invoked following a staged file transmission in both the source and target SOLVE management services. It is used to reconstruct (if necessary) the prefix allocated to staging datasets so that the reload operation on the target dataset and deletion of the staging file can proceed. It may also be used to return SOLVE management services console user ID when user notification is required after a remote FTS request has completed. The NCL procedure is driven with two passed parameters and must return the relevant NCL variable as described in

the comments within the procedure. The second parameter passed is the TSO user ID of the user who requested the transmission which may be useful in determining the dataset prefix value to be returned. The following variables can be requested:

PREFIX

Is the staging dataset prefix for this user. See the section on *Managing Temporary Datasets* below for full details on how this should be used.

SYSOPER

Is the name of the SOLVE management services console user ID in this system that can be used to establish a ROF session with the SOLVE management services which is running on the system where the TSO user originally requested the transfer. Unless FTS completion notification is required after requesting a transmission from a remote system then this parameter need not be set. (Notification following a transfer from the local system is always available). The console user ID returned in this parameter must be able to issue the ROUTE command, to connect to the SOLVE management services where the TSO user originated the request, and on that system issue an OPSYS command to send the necessary notification to the user.

Operational Considerations

Invoking the Dialogue

The dialogue is intended to run as a ISPF Dialogue application. This means it has its own operational shared variable pool, and its own profile dataset member in which information is stored across application sessions.

The dialogue application can be tested from option 7.1 of ISPF (by invoking procedure NMFTS and specifying a NEWPOOL ID of NMFT for the call) but it should ultimately be invoked from some convenient user menu (for example, a tailored primary ISPF menu). The following dialogue statement can be used to gain entry to the dialogue system from such a menu:

```
ISPEXEC SELECT CMD(NMFTS) NEWAPPL(NMFT)
```

After being invoked for the first time the dialogue application will create a member NMFTPROF in the user's ISPF profile dataset on exit.

FTS Transmission Request Names

Each transmission request in a given SOLVE management services must have a unique name. The dialogue uses the TSO user's user ID plus 1 character (A to Z) in order to construct both a valid and unique transmission name. This allows the user the potential of having up to 26 requests outstanding in a given SOLVE management services. The dialogue is responsible for determining the next character to be used in constructing the transmission name. This name is returned in a message to the user after the request is successfully processed and the transmission queued.

Managing Temporary Datasets

Dialogue procedures are designed to eliminate temporary datasets, allocated for staged file transmissions, as soon as the request is complete. However, various errors are possible (such as link failure etc.) which result in such datasets not being deleted by the usual system processes.

Since every user of the ISPF dialogue will not be aware of the various mechanisms involved in scheduling the transmission, it is assumed that the installation will need to perform some regular house-keeping procedures to delete any unwanted datasets.

This requires knowledge of the names allocated to temporary datasets. The installation has some control over these names by setting the variables &LCLPREF and &REMPREF as described in the above section. This prefix is used to construct the temporary dataset names used by the dialogue in the following manner.

The staging dataset name used in the sending system is:

```
&LCLPREF . &FTSNM . NMSTGO
```

The staging dataset name used in the receiving system is:

```
&REMPREF . &FTSNM . NMSTGI
```

The IEBCOPY control file name allocated in the sending system is:

```
&LCLPREF . &SYSUID . SYSIN
```

The variable &SYSUID is merely the TSO user ID of the person using the dialogue and is a standard dialogue system variable. The variable &FTSNM is the TSO user ID (&SYSUID) plus 1 character (A to Z) which is the transmission name constructed by the dialogue to uniquely identify this request.

Only one IEBCOPY control file is ever required and is deleted after the output staging dataset has been created. The staging dataset names are tied to the transmission request names for ease of re-constructing at the receiving end and also as a convenient means to make the name unique.

Depending upon the exact nature of the staged transmission, particular NCL procedures are invoked upon its successful completion in both the sending and receiving SOLVE management services. In the sending system the procedure \$@DELETE is always invoked and the value of &LCLPREFIX, defined by the installation as the local staging dataset prefix, is passed as a parameter to this procedure. It deletes the staging dataset on the local system and notifies the TSO user of the successful transmission. In the receiving system the prefix cannot be carried as a parameter and must be determined by calling the NCL procedure \$@NMFTS. When determining what values to use for the &REMPREF variable defined in the dialogue procedure NMFTSID it should be remembered that procedure \$@NMFTS, executing in the remote SOLVE management services, must be able to reconstruct such a prefix given only the original TSO user ID of the requestor. It returns this value in &PREFIX to the calling NCL procedure (\$@COPY, \$@REPRO or \$@E) which then constructs the entire staging dataset name from which to reload the target dataset and then delete.

Error Recovery

When problems do occur it is not always possible to notify users that their file transmissions have failed. Those users who are familiar SOLVE management services users may log onto the source SOLVE system and determine the cause of the FTS failure through standard FTS panels. Once the error is isolated the FTS request may be restarted if possible, or else deleted and a new request issued. Remember that for staged transmissions this may result in datasets not being deleted as described above.

For users who are unfamiliar with SOLVE management services, an installation may need to allocate such tasks to operators with FTS facility experience.

Note: NCL procedures executed after a successful file transmission write messages to the SOLVE management services activity log which may be another useful source of information.

Note: More complete notification techniques, following either successful or unsuccessful file transmissions, could be implemented by an installation interested in pursuing this by trapping the relevant messages (both standard FTS messages and the messages issued by the NCL procedures executed at successful completion of the transmission) in LOGPROC.

An installation can implement new NMCMD sub-commands which allow users to track transmission requests and delete unwanted requests and datasets.

Using the Dialogue

It is assumed that users familiar with ISPF will have no difficulty using the system, as the general panel presentation and flow are similar to other ISPF options.

On entry, a FROM panel is presented where details of the type of transmission request, the dataset the user desires to transmit and the system on which it resides, are requested. Valid options are:

C

To perform a *direct* copy from the source dataset to the target dataset. The source dataset may be a sequential file or a PDS member. If the source system is the local system then the source dataset may specify a PDS and a selection list will be provided if no member name was supplied. If the source system is a remote system and no member name is supplied then the source dataset is assumed to be a sequential file.

M

To perform a *direct* copy specifying multiple members of a PDS. The source dataset must be a PDS on the system specified. The system may be either local or remote. A panel will prompt for the member names to be transmitted once TO details have been supplied.

S

To perform a *staged* copy from a local PDS or VSAM dataset only. For a PDS dataset a member name may be supplied, specified as asterisk (*) to indicate all members, or left blank to request a member selection list.

The *from* system is checked against the valid names supplied in the tailoring procedure NMFTSID. If it is recognised as the local system then the dataset is checked to discover its organisation (DSORG) and allocation details. This information is used for cross-editing the request. In the event that a NEW target dataset is requested, such information is used to prime the panel prompting for the new dataset allocation details.

Following the entry of *from* details a TO panel is presented which requests the information about the target dataset.

If option M was selected then both the source and target datasets must be PDSs and no member names can be supplied as they are entered on a subsequent panel.

If option C was selected and the *from* system was determined to be the local system, then the target dataset is edited as follows:

- ▶ The source dataset will have been found to be either a PDS or a sequential file, otherwise the request would have been rejected. If it is sequential then the target dataset may be either a single PDS member or a sequential file (which is assumed if no member name was specified).
- ▶ If the source dataset was a PDS and no member was specified then a selection list will be presented following the TO panel and hence no member name is allowed for the target dataset which is assumed to be a PDS anyway. If the source dataset was a PDS and a member name was specified then the target dataset may be either another PDS member, or a sequential file. When no member is supplied in the target dataset name, then if the area of the panel specifying an ISPF LIBRARY is used to describe the dataset it is assumed to be a PDS and the member supplied on the FROM panel is assumed to be the target member for the *to* dataset. Otherwise the dataset is assumed to be a sequential file.

If option C was selected and the *from* system was determined to be a remote system, then the target dataset is edited as follows:

- ▶ If a member name is specified on the FROM panel, the dataset is assumed to be a PDS, otherwise it is assumed to be a sequential file. If a member is specified on the target dataset it too is assumed to be a PDS.
- ▶ When no member is supplied in the target dataset name but one was specified in the source dataset name, then if the area of the panel specifying an ISPF LIBRARY is used to describe the dataset it is assumed to be a PDS and the member supplied on the "FROM" panel is assumed to be the target member for the *to* dataset. Otherwise the dataset is assumed to be a sequential file.

The target dataset may be specified as NEW for a direct copy or staged VSAM transfer only. For a direct copy (option C or M) another panel requesting allocation details is displayed.

For a staged copy of a VSAM dataset a panel will prompt for information on whether VSAM record replacement is allowed, and for a limiting range of keys which should be copied (applicable to KSDS datasets only). If the target VSAM dataset was specified as NEW then a panel is displayed requesting confirmation of the transmission before a temporary export is performed, the dataset transmitted and an import performed at the receiving end. This is subject to any restrictions imposed by the IDCAMS EXPORT/IMPORT processing.

For options C and S with a source dataset which is a PDS on the local system, a member selection list is provided where no member name was specified on the FROM panel. Users may scroll through the selection list in the usual manner until all selections have been made. Members may be selected with an S to copy the member with the same name, or an R to copy the member in which case a new name must be entered. Each entry may only be selected once, any subsequent selections of the same member being ignored. Pressing the END key terminates the selection processing and cancels the request if no selections were made. To cancel the operation having made one or more selections use the CAN or CANCEL primary line command.

Once all the information required to define the transmission has been obtained a panel is displayed requesting the user to wait while the transmission is defined and scheduled. During this time the records selected for a staged transmission are unloaded into the staging file. Following this (or immediately for the direct copy options), the NMCMD TSO command processor is invoked to send the FTS request to the appropriate SOLVE management services.

Any error messages occurring during the dialogue, or any informational messages returned by SOLVE management services following the successful scheduling of the request, are then displayed. Finally control is returned to the FROM panel where the process can be repeated, or the END key pressed to exit.

Customer Support Services

Sterling Software provides a variety of customer support services to ensure that you get the most out of your Sterling Software products. The sections that follow provide you with information to make it easier for you to communicate with our Customer Services staff. The sections include our telephone hotline and fax numbers, Support Centre addresses, and outline basic information you should have on hand when calling your local Support Centre for assistance. If you have questions other than product support, our Technical Support Representatives can also help you get the information you need.

If you have questions about billing, new product orders, or the Product Support and Enhancement Agreement, contact your local Sterling Software representative.

Technical Support

If you encounter a problem with one of our products that you cannot resolve by reading and following the online help or the documentation, call your local Sterling Software Support Centre. Our Technical Support Representatives are available to answer your questions about Sterling Software systems and network management software.

To receive prompt service, use the following hotline numbers for the countries and products listed. If you are outside these countries, contact your local Sterling Software representative for help.

In most regions, you can call the hotline number outside of prime support hours if your problem is critical and cannot wait until the next business day. We have a Technical Support Representative on call 24 hours a day, 365 days a year for such emergencies.

If the Technical Support Representative asks you to send information or documentation regarding your problem to the Support Centre, use your local Support Centre's fax number or mailing address.

You can help the Technical Support Representative solve your problem more quickly if you have the following information at hand before you call:

- ▶ Your system model, make, and serial number
- ▶ Your problem tracking number and customer number, as these apply
- ▶ Your operating system, release level, and maintenance level
- ▶ Release and maintenance level of the product or component causing the problem
- ▶ Final error messages, especially the message numbers, that the problem caused
- ▶ All product manuals for easy reference

If all Technical Support Representatives are busy and you have to leave a message, remember to provide your name, your company name, phone number, call incident or problem number if you have one, and a brief description of the problem. You can also leave other relevant information such as how to reach you if you are not available at the stated phone number. A representative will return your call as soon as possible.

Local Support Centres

Sterling Software Support Centres are listed alphabetically by region below. When calling for technical assistance, please use the Support Centre in your region. If you are outside these countries, contact your local Sterling Software representative for help.

Europe

France

Direct Support Hotline: 33 1 47 67 40 40
Prime Support Hours: Normal business hours
Monday through Friday

Fax: 33 1 47 67 40 41

Mailing Address: Paris Customer Support Centre
Sterling Software
70 Avenue du President Wilson
92800 – Puteaux

Germany

Direct Support Hotline: 49 6102 709 0
Support Hours: Normal business hours
Monday through Friday

Fax: 49 6102 709 111

Mailing Address: Technischer Kunden–Service (TKS)
Sterling Software GmbH
Schleussnerstrasse 54
D–63263 Neu–Isenburg

Italy

Direct Support Hotline: 39 11 771 4095
Support Hours: Normal business hours
Monday through Friday

Fax: 39 11 743 878

Mailing Address: Sterling Software Srl
Corso Svizzera 185
10149 Torino

Norway

Direct Support Hotline: 47 22 65 10 52
Prime Support Hours: Normal business hours
Monday through Friday

Fax: 47 22 64 66 79

Mailing Address: Sterling Software
Brynsveien 13
Postboks 6392 Etterstad
N–0604 Oslo

Sweden

Direct Support Hotline: 46 (0)8 752 7900
Prime Support Hours: Normal business hours
Monday through Friday

Fax: 46 (0)8 752 7337

Mailing Address: Sterling Software
Borgarfjordsgatan 16
S-164 40 Kista

United Kingdom

Direct Support Hotline: 44 734 605609
Prime Support Hours: Normal business hours
Monday through Friday

Fax: 44 734 605540

Mailing Address: Reading Customer Support Centre
Sterling Software
75 London Road
Reading
Berkshire RG1 5BS

North America**Canada**

Contact US Support Centre

United States

Direct Support Hotline:
Metro Washington 703 264 8334
Continental US 1 800 663 6529

Prime Support Hours: 8:30 a.m. to 5:30 p.m. EST
Monday through Friday

Fax: 703 264 0751

Mailing Address: Reston Customer Support Centre
Systems Management Division
Sterling Software
1800 Alexander Bell Drive
Reston, VA 22091

Pacific Rim

Australia or New Zealand

Direct Support Hotline: 02 975 8469
Prime Support Hours: 8:30 a.m. to 5:00 p.m. EST
Monday through Friday

Fax: 02 975 5245

Mailing Address: Sydney Customer Support Centre
Sterling Software
Forest Corporate Park
28 Rodborough Road
Frenchs Forest NSW 2086

Japan

Direct Support Hotline: 03 5472 5531
Prime Support Hours: 9:00 a.m. to 5:30 p.m.
Monday through Friday

Fax: 03 5472 5530

Mailing Address: Sterling Software
Shibakoen Takahashi Building
8 – 12 Shibakoen 1 – Chome,
Minato-ku, Tokyo 105

Singapore

Direct Support Hotline: 65 732 9996
Prime Support Hours: Normal business hours
Monday through Friday

Fax: 65 733 4358

Mailing Address: Sterling Software
501 Orchard Road
#15–03 Lane Crawford Place
Singapore 0923

User Conferences

Sterling Software holds annual user conferences at locations throughout Europe, North America, and the Pacific Rim. These conferences give you an opportunity to learn from the experiences and expertise of other Sterling Software customers and to share some of your product knowledge with them. Each conference includes technical sessions, user presentations, roundtable discussions, and time for informal exchanges between customers and Sterling Software employees.

For more information about our user conferences, contact your local Sterling Software representative.

Installation Considerations

Installing EIP Components

Following initial installation of either the EIP feature or FTS feature, further steps may be required to ensure that the required EIP components are in libraries which are accessible to the requesting users and programs. This appendix outlines the general installation considerations for EIP followed by the further considerations for users of the FTS External Interface facilities.

EIP Load Libraries

OS Load Libraries

The load modules which comprise the various EIP components are installed into target load libraries. By including such libraries in suitable STEPLIBs or the system LINKLST these target libraries can be used directly by EIP users. Otherwise some load modules will need to be moved from the installation load libraries to the appropriate system libraries.

For TSO users this means that load modules NMLOGON, NMCMD and NMV3270I (found in the installed CMDLIB), should be placed in either a TSO command library or system LINKLST library. See your *SOLVE Management Services Installation Instructions* for MVS and equivalent operating systems.

Module NMBCI, found in the base installation load library, can be placed in any convenient library accessible to batch programs (for example, the system LINKLST). Module NMV3270I is loaded by NMBCI and must be placed in a library accessible to it (for example, in the system LINKLST).

VSE Core-Image Libraries

For VSE users the modules NMBCI and NMV3270I are distributed in the installation core-image library and may be placed in any convenient core-image library.

EIP NCL Procedures

For OS users, the distributed NCL procedures and panels are installed into the target EXEC and PANELS datasets respectively. If these are not appropriate then they should subsequently be moved to the desired NCL execution libraries and panels datasets for each SOLVE management services which will require their use.

For VSE users, the NCL procedures and panels are installed into the N.SUBLIB source statement library and panels file respectively.

The NMLOGON TSO command processor, and the NMV3270I service module, have no specific NCL requirements, but both NMCMD and NMBCI are designed to perform their processing requirements within User Services. The standard User Services procedure named \$USERSER is used by NMCMD, with additional procedures \$USERBCI being required for the BCI, and \$USERTSO being required for the TSO Command Interface. Both of these interfaces also require panel \$USV3270.

VTAM APPL Definitions

VTAM APPL definitions must be updated to include the entries required by the EIP components. One set of definitions should be included on each system which is to use these interfaces.

The default names used are NMBCI001, NMBCI002, and so on, for the BCI, and NMTSO001, NMTSO002, and so on, for both the TSO Full-Screen Interface and the TSO Command Interface. The entries required should be defined as follows:

```
applname APPL EAS=1,ACBNAME=NMBCInnn,
```

and

```
applname APPL EAS=1,ACBNAME=NMTSOnnn
```

Note: It is possible from all interfaces to specify directly the name of the ACB to be used for the session. However, coding the above names is convenient as these default definition names will be found by the interface programs if a specific ACB name is not requested.

SOLVE Management Services User IDs

When using any of the interfaces supplied with EIP a standard logon is performed on behalf of the user. This of course requires a valid *user ID* and *password* which must be known to SOLVE management services. Although in some interfaces this logon is completely transparent to the user, internally to SOLVE management services there is no difference and the logon will be rejected if the user ID or password is invalid, or if the user is already logged on from some other device.

For the BCI we recommend that special user IDs be set up, thus preventing logon failures due to concurrent terminal and batch usage. Any additional SOLVE management services user IDs added for this purpose should have their initial passwords changed by first logging on at a standard terminal.

All users should note that it is a standard security feature of UAMS component of SOLVE management services, and usually of installation security exits in general, that passwords must be changed regularly. Password expiry can cause failure to connect through some interfaces and it is the installation's responsibility to ensure that passwords are reset online as required, and that where necessary any BCI input and

the BCI command exit is suitably amended or protected against such changes.

Users who are to use the NMCMD interface should be allocated a User Services procedure name of \$USERSER, while NMBCI users should be allocated \$USERBCI.

For TSO users, the installation should also consider allowing TSO AUTOLOGON which provides TSO users with the ability to logon to SOLVE management services directly from their TSO session without having to re-enter their password. This option is available when defining the UAMS definition for a user. Note that if the logon request comes from anywhere other than from an on-line TSO session then the user ID and password will always be required.

The Sample BCI Command Exit

A sample BCI command exit program, member BCIEXITC, is also supplied in the SOLVE management services installation library. Those installations intending to use a BCI command exit can modify this sample program to implement their own requirements. The exit must be assembled and link-edited into a library accessible to NMBCI before it can be used. The BCI will load the command exit module when it is activated.

The supplied command exit, BCIEXITC, will process commands as follows:

```
CATALS          command is ignored
BKEND           command is ignored

LOGON           will generate the following
BCILOGON command:-
                BCILOGON NM2 USER BCI1 PASSWORD BCI

MYSHOW         will generate the following commands:-
                SHOW PARM
                SHOW USERS
                SHOW SESSIONS
```

Testing the Interfaces

Having reviewed the above installation considerations you should test the TSO Full-Screen Interface, TSO Command Interface, and BCI, as applicable to your operating system, to verify the installation processes. See the beginning chapters of this manual for details on how to invoke the various interfaces.

The FTS External Interfaces

The remainder of this appendix documents the further steps to be considered when using the FTS External Interfaces. If the FTS feature is not installed in your system, or you do not intend to use these interfaces, then there is no need to read any further.

FTS NCL Interface

The FTS NCL procedures and full-screen panels are supplied with the FTS feature in the installation EXEC and PANELS datasets respectively. All procedure and panel names start with the characters \$FTS.

A SYSPARMS SYSUDB=YES command must be entered before using any of the procedures, either from OCS or, more conveniently, in the NMINIT procedure during initialisation. This permits the NCL procedures to access the SOLVE management services VFS file so that it may add FTS definitions.

Testing the FTS NCL Interface Procedures

To test the procedures as supplied, one of three options can be taken:

- ▶ Execute the following command from OCS mode:

```
FSPROC $FTSMENU
```
- ▶ Modify a user ID so that the \$FTSMENU procedure is executed when the User Services option is requested from the SOLVE management services Primary Menu.
- ▶ Change an existing User Services system so that a new option invokes the \$FTSMENU procedure.

To use the TRANSMIT A FILE option, you must be authorised to use FTS and to request private definitions, and be authorised to transmit the dataset in question (by RACF or ACF2 if necessary).

User Services Invocation

To implement an ad hoc transmission capability based on \$FTSDEMO, and have procedures invoked from User Services, you must be authorised for both FTS and User Services. If you are authorised for more than one SOLVE management services function, you will normally be presented with the SOLVE : Primary Menu when you log on.

If desirable, a user can be placed directly into the User Services panel at logon time. If logging on from the EASINET facility, this is accomplished by having the EASINET NCL procedure pass user ID, password and an initial menu selection of U as logon user data. For example:

```
&LOGON NM USER254 MYPW U
```

If logging on via the TSO Full-Screen Interface, this may be accomplished by passing the initial menu selection on the NMLOGON command, which can in turn be invoked from a TSO CLIST. Users defined to SOLVE management services with the TSO AUTOLOGON attribute are not required to supply their user ID and password but are automatically logged on to SOLVE management services. For instance, a TSO CLIST called FTS might execute the command:

```
NMLOGON APPL(NMFTS) MENU(U)
```

In both the above examples, the User Services procedure is invoked and can display menus, invoke processing, and so on. The &QEXIT statement can then be used from within the NCL procedure to provide an express logoff which bypasses the Primary Menu display when exiting.

FTS TSO Interface

The NCL procedures required for the FTS functions of NMCMD are supplied with the FTS feature in the installation EXEC library and are prefixed by \$US...

Installation considerations for the load modules NMCMD and NMV3270I used by this interface are discussed above.

Testing the FTS TSO Interface Procedures

Firstly ensure that the TSO Command Interface is operational and that all requirements specified for the FTS NCL Interface have been met. Then the FTS sub-commands of the NMCMD TSO command processor, as described in the *FTS TSO Interface* chapter, can be tested.

FTS ISPF Dialogue Interface

The various components comprising the FTS ISPF Dialogue Interface are found in several installation libraries. Ensure that all installation considerations previously described for the FTS TSO Interface have been addressed before continuing.

The TSO Environment

The following installed datasets contain components which must be accessible from ISPF for the dialogue users. This may be achieved either by concatenating the datasets to the appropriate DD statement in user ISPF procedures, or by copying the contents into existing datasets:

- ▶ Installation CMDLIB dataset contains load modules called from the ISPF Dialogue, NMFTDSRG and NMFTMSEL.
- ▶ Installation ISPCLIB contains ISPF Dialogue procedures.
- ▶ Installation ISPLIB contains ISPF Dialogue panels.
- ▶ Installation ISPLMLIB contains message members for the @note = ISPF Dialogue.

Non-TSO/E Considerations

The ISPF Dialogue invokes the IEBCOPY system utility under certain circumstances. This program must run authorised which is achieved by the use of the TSOEXEC invocation under TSO/E. However, this is not valid in a non-TSO/E environment and the ISPF dialogue must be changed if this facility is being used.

Many installations affected may have an alternate program called SPFCOPY (or similar) available which can perform an authorised copy operation. (If not, such a sample program can be found on the system SAMPLIB from where it must be assembled and linked with the RENT attribute to the system LPALIB or LINKLIB. It may additionally be required to install the SVC235, called ISR235 on the system SAMPLIB, by linking it with the RENT and REFR attributes into the system LPALIB. This SVC is used to gain authorisation and the entire process is more fully documented in the appropriate ISPF Installation and Customisation manual).

Having identified or installed an SPFCOPY utility, the dialogue procedure named NMFTSS must be changed. Locate the statement

```
TSOEXEC CALL `SYS1.LINKLIB(IEBCOPY)`
```

and change it to call the appropriate program, for example:

```
CALL `SYS1.LINKLIB(SPFCOPY)`
```

The dialogue procedures will then operate as described.

SOLVE Management Services Environment

Various NCL procedures in the installation EXEC library are invoked by FTS completion From and To commands. These members have names beginning with the characters \$@ and must be available on each system which is to be used for file transmissions. If the installed EXEC library is unsuitable, then they must be copied to another NCL library.

In addition, the SOLVE management services utilities UTIL0007 and UTIL0009 within the installation load library, are invoked from the NCL procedures mentioned above and hence must be in load libraries accessible to SOLVE management services.

Use of FROM and TO Commands

The FTS feature allows commands to be executed following a successful transmission. One command can be executed in the sending or From end, and one in the receiving or To end. Since commands executed in this manner have no command authority checking, this facility is restricted to FTS *system* transmission requests by the FTS feature.

The FTS ISPF Dialogue Interface uses the FROM and TO command capability to execute NCL procedures after the transmission completes. These are used to notify the TSO user of the successful transmission completion and also to drive further procedures for staged transmissions which reload the target datasets from the *staging files* and delete the staging files.

Because of these requirements, the dialogue uses FTS system definitions and requests. Hence dialogue users require FTS *system request* authority to use the dialogue. However, FTS *system definition* authority is not required as the definition is created on the VFS file by NCL procedures, so is not validated by FTS itself.

If an installation requires that the ISPF dialogue only use FTS *private* definitions and requests then the NCL procedure \$USTSFTS must be changed to allow the specification of FROM and TO commands for such requests.

One further restriction of these FTS transmission completion commands is that the command length is limited to 40 characters. For some functions the ISPF Dialogue requires a completion NCL procedure to be driven with two parameters. For example:

```
$@L &FTSNM &DSNAME
```

would be used to reload a PDS after a staged file transmission. Dialogue variable &FTSNM is the transmission request name, consisting of the user ID plus a character, and hence can be up to 8 characters in length. This reduces the maximum length of the fully qualified dataset name (the variable &DSNAME which contains the name of the actual target dataset to be reloaded) in the worst case to 27 characters.

Note: The FROM and TO commands used by the dialogue do not specify the EXEC command, only the procedure name plus its parameters. This implies that the SYSPARMS AUTOEXEC=YES option is in effect in the SOLVE systems concerned.

Staged PDS Transmission Requirements

Following a staged transmission of one or more PDS members an NCL procedure will be invoked in the receiving system to perform the reload operation. Procedure \$@L is used to load the target PDS with *no* member replacement, while procedure \$@M is used where member replacement is allowed.

Both procedures use another procedure named \$@COPY which performs the required dataset allocations and calls UTIL0009 to attach IEBCOPY. Only the staging and target datasets are dynamically allocated. The other DD statements required by IEBCOPY must be pre-allocated.

The DD names required by UTIL0009, and their purpose is as follows:

IEBCSOD

This is the required DD name for IEBCOPY output (i.e. the usual SYSPRINT file). It may reference any suitable SYSOUT dataset or a dummy dataset.

IEBCSILN

This DD name is used as the IEBCOPY SYSIN file for reload operations where member replacement is not permitted. The IEBCSILN DD should refer to a suitable dataset or PDS member which contains the single statement:

```
COPY OUTDD=IEBCDD2 , INDD=IEBCDD1
```

IEBCSILR

This DD name is used as the IEBCOPY SYSIN file for reload operations where member replacement is permitted. The IEBCSILR DD should refer to a suitable dataset or PDS member which contains the single statement:

```
COPY OUTDD=IEBCDD2 , INDD=( ( IEBCDD1 , R ) )
```

UTIL0009 overrides the standard IEBCOPY DD name defaults to allow the FTS completion procedures to select the DD name which contains the appropriate IEBCOPY control statement.

Any or all of the above datasets may be allocated dynamically after SOLVE management services has started, by using the ALLOCATE command. However, we recommend that they be included in the SOLVE management services execution JCL, for convenience.

Security Considerations

NCL procedures do not currently have access to all user ID information, and so the supplied FTS NCL procedures cannot determine whether a user is authorised to read or modify a particular transmission definition. For example, determining whether the user is a private or system user, what masking applies, and so on.

For this reason, the supplied FTS NCL Interface procedures (by default) render a definition as a *private definition* when writing it to VFS, as this is the most secure option. However, as discussed above, the FTS ISPF Dialogue Interface requires the use of FTS From and To commands and hence uses FTS *system definitions* to perform the transmission requests.

So that the FTS NCL Interface can mirror the base FTS processing, the procedures can be modified to reference a UDB containing user ID information that can only be supplied by selected security personnel. Alternatively, the &SECCALL NCL statement can be used to converse with the installation security exit.

You should also consider implementing of an &FILE security exit (NCLEX01) to verify that the VFS dataset is only referenced by selected personnel, or is only referenced from selected procedures.

FTS procedures may themselves be placed in a secure dataset, with that dataset being placed first in the NCL library DD concatenations (OS only).

Glossary

This glossary defines the terms and abbreviations commonly used with SOLVE management services.

It also includes references to terms used in an IBM environment and any equivalent FUJITSU terms.

3270 VDU terminal

An IBM video display terminal. This is often used to refer to the entire range of 3270 terminals. When followed by a number (for example, 3270-5), a specific model is intended.

370/390

This is an abbreviation for IBM's System 370 or S/370 architecture. It is often used to indicate any mainframe CPU that implements this architecture.

3705/3725/3745

An IBM front end communications processor (The Fujitsu equivalent is a CCP or 2806).

9526

A Fujitsu video display terminal.

ACB (Access method Control Block)

A control block that links an application program to an access method such as IBM's VTAM or VSAM.

ACB-sharing

MAI's ability to use a single VTAM ACB for multiple sessions.

Access Security Exit

An installation-provided routine that may be used to replace the SOLVE management services UAMS functions, partially or completely, allowing logon, logoff, and password maintenance requests to be passed to an external security system.

ACF/VTAM (Advanced Communication Facility/VTAM)
IBM's product implementation of SNA's SSCP or CP.

Activity Log

A system-maintained log that records all important activity for use in later problem determination.

Advanced Program to Program Communications (APPC)

An IBM-defined application level protocol which makes use of SNA's LU 6.2.

AIM

Fujitsu's main database/data communications system.

Alternate index

An alternative view of the data contained within a VSAM keyed dataset. The alternate index allows data to be retrieved using an alternate key in addition to the usual access through the primary key.

AOM (Advanced Operation Management)

A facility of SOLVE management services that manages and controls local and remote operating systems.

AOMPROC

The name given to an NCL procedure used to intercept messages from the screening table component of AOM to provide extended message processing.

APF (Authorised Program Facility)

Describes the special authorisation level required within the operating system for certain applications.

APPC (Advanced Program to Program Communications)

An IBM-defined application level protocol which makes use of SNA's LU 6.2.

APPL

A VTAM term used to describe the definition that allows an application to use VTAM facilities.

Application Plan

A DB2 term that refers to the control structure used by the database to allocate resources and execute SQL statements.

ASN.1

Abstract Syntax Notation One, defined by ISO 8824, is an abstract syntax used to describe data structures. It is used by Mapping Services to define data structures within SOLVE management services.

Authorised Program Facility (APF)

Describes the special authorisation level required within the operating system for certain applications.

Authorisation ID

A DB2 term that refers to the ID used by DB2 to control access to a database.

BER (Basic Encoding Rules)

The transfer syntax used by Mapping Services to serialise data for transmission. It is defined by ISO 8825.

BIND

1. A VTAM term describing the action of logically linking one network resource with another network resource.
2. A DB2 term for the process that connects together the application plan and the Database Request Module (DBRM) used by an application.

Broadcast Services

Broadcast Services controls the sending of messages throughout SOLVE systems.

CAF (Call Attach Facility)

A connection technique provided by IBM and used by the EDBS facility of SOLVE management services, to communicate with DB2.

Checkpoint

Refers to a point of synchronisation in processing where a unit of work is complete, or partially complete, such as where data is recorded for restart purposes. A point at which information about the status of transmission can be recorded so that it can be restarted later.

CICS (Customer Information Control System)

An IBM licensed program that enables transactions entered at remote terminals to be processed concurrently by user-written application programs.

Client

A functional unit that receives shared services from a server.

CNM (Communications Network Management)

IBM term for its SNA management facilities.

CNMPROC

The name given to an NCL procedure used to intercept CNM records received across the VTAM CNM interface by the NEWS component of SOLVE:Netmaster.

Command Partition

A term associated with NPF that describes the group of network resources a user ID is authorised to reference with VTAM commands.

Configuration Management

An ISO/OSI classification of management functions that apply to the ability to set or change operating parameters of the system, to collect and distribute information on their status, to associate names with the entities, and to change the system configuration.

Control Member

A term associated with NPF that describes the list of resource table names applying to a user ID. This control member is referenced in the definition of USERID.

Cross-domain Resource

A VTAM term describing the definition of a Network resource that is owned by a VTAM in another domain.

DBCS (Double Byte Character String)

Refers to a mode of representation of data where each byte of data requires 16 bits rather than 8 bits as required by Single Byte Character String. DBCS is used for the implementation of languages such as Kanji, which is used in Japan.

Deferred Write

A performance option for use with UDBs to minimise I/O by deferring the writing of records.

DEFLOGON

The term used to describe an application entry path to be supported by the EASINET and MAI-FS features. The DEFLOGON command is used to define these paths and their associated text strings.

Dependent Processing Environment

An NCL processing environment which is a child process to another NCL process and hence has its output delivered as input to some other NCL procedure.

DOM (Delete Operator Message)

A non-roll deletable message (NRD) can be deleted from a window only when a DOM is issued.

Domain

1. An SNA term describing a domain that consists of the set of SNA resources controlled by one common control point called an SSCP. In terms of implementation, an SSCP is the host access method (VTAM). An SNA network consists of one or more domains.
2. A VTAM term that describes a logical division of a network. Networks are divided into domains that are associated with the way they are controlled.

Domain ID

Term for a 1–4 character mnemonic used as a unique identifier for a SOLVE system.

Dynamic Allocation

Assignment of datasets to a program at the time the program is executed rather than at the time the job is started.

EASINET

The name of the set of SOLVE management services functions available from SOLVE:Access. EASINET allows idle terminals to be brought under the control of SOLVE management services and to be operated by installation-written NCL procedures that can provide a wide range of *front end* facilities to end-users of the network.

ER (Explicit Route)

The physical path between two network nodes. (SNA)

ESDS (Entry Sequenced Data Set)

A non-keyed VSAM dataset whose records are stored and retrieved in sequential order, and new records added to the end of the data set.

ESF (Expert Systems Foundation)

A rule-based facility that automates the handling of messages captured by AOM.

Exception

The result of a service request that did not complete successfully. See *Reply* and *Response*.

Exit

An installation-written routine that can be driven from a point within a program to provide data to the program, or perform additional processing relevant to that installation's specific requirements.

Extended Datastream

A 3270 datastream containing fields that utilise colour and extended highlighting capabilities of the terminal.

External Database Connection

A term used to refer to connections from SOLVE management services to the database which use the Call Attach Facility (CAF) to connect to DB2. NCL connections are connections to DB2 that utilise the management services connection. See also *Management Services Connection* and *NCL Connection*.

External Database ID

The ID of the external database as it is known to the operating system.

Fault Management

An ISO/OSI classification of management functions relating to the detection, collection, diagnosis, testing, correction, and prevention of network and system faults and error conditions.

FID

A File ID under DOS.

FSP

A Fujitsu operating system.

Function Key

A key on a terminal's keyboard which causes a panel to be completed. In the case of 3270, this term also applies to the ENTER key (the same behaviour can be achieved on a 6530). Previously called Program Function (or PF) keys.

IMS (Information Management System)

IBM's database/data communication (DB/DC) system that can manage complex databases and networks.

Initiator

The component of FTS which schedules transmission of a dataset.

INMC (Inter-Management Services Connection)

This facility allows SOLVE systems running in a network containing multiple CPUs to communicate with each other, providing general-purpose data transfer between CPUs within the network.

INMC/EF (INMC/Extended Function)

INMC/EF provides the capability for up to sixteen sessions between any pair of SOLVE systems. In appropriate systems, these sessions can traverse different physical network paths, thus increasing throughput. This component also provides additional link security and management facilities.

In releases prior to Version 3.0, INMC/EF was a separate component. At release 3.0 it became a part of INMC.

Inter-Management Services Connection (INMC)
See *INMC (Inter-Management Services Connection)*.

I/O
Input/Output

ISPF (Interactive System Productivity Facility)
An IBM product that is the functional equivalent of Fujitsu's PFD.

KSDS (Key Sequenced Data Set)
A VSAM dataset whose records are directly accessed by a user-supplied key.

Link
A term used to describe a logical connection between two or more SOLVE systems. See also *INMC (Inter-Management Services Connection)*.

Logical Unit (LU)
The point of access for any user to an SNA network. See also *LU*.

LOGMODE
A VTAM term used to describe a table entry that defines the characteristics and protocols of a terminal.

Logoff
A request by an LU that it be disconnected from a VTAM application program.

Logon
A request by an LU that it be connected to a VTAM application program.

Logon-path
A path through which users of EASINET and MAI-FS gain access to other VTAM application programs. Paths are defined and controlled by the DEFLOGON command.

LOGPROC
The name given to an NCL procedure used to intercept messages destined for the SOLVE management services activity log.

LSR (Local Shared Resources)
A technique for buffering I/O to VSAM files called LSR pools. NCL supports this type of processing for User Databases (UDBs).

LU (Logical Unit)
SNA introduced the concept of the logical unit (LU). The LU is a type of SNA network-addressable unit (NAU) that provides protocols for end users to gain access to the network and to the functional components of the LUs. See also *Logical Unit*.

LU0

An unconstrained SNA protocol that allows implementors to select any set of available protocol rules, as long as the two LUs are able to communicate with each other successfully according to the rules chosen. Therefore, all LU types are an implementation of LU Type 0.

LU1

A line-by-line or typewriter type terminal (for example 3767, 3770), using SNA protocols.

LU2

A 3270 type terminal using SNA protocols.

LU3

LU Type 3 was implemented to support printers with a different data stream format. LU Type 3 is used by printers attached to an IBM display cluster controller.

LU4

LU Type 4 was implemented so that office system products could transfer documents. LU Type 4 is used by banking devices.

LU6.2

A protocol that serves as a port into an SNA network. LU6.2 defines a specific set of services, protocols, and formats for communication between logical processors. LU6.2 provides presentation services for presentation of data to the end user, transaction services for performing transaction processing on behalf of the end user and LU services for managing the resources of the LU.

LU7

An SNA protocol that is used by word-processing devices.

Macro

A single source language statement that expands into a series of statements when the macro is assembled or compiled.

MAI (Multiple Application Interface)

A facility of SOLVE:Access which is used to provide sessions with any number of other VTAM application programs from one SOLVE terminal.

MAI/EF (MAI/Extended Function)

A SOLVE:Access facility which provides an extension to NCL to support session scripts which allow an NCL procedure to control the application and terminal session flow. MAI/EF also includes the Session Replay Facility (SRF) which allows recording and replay of session scenarios.

MAI-FS (MAI/Full Screen)

A SOLVE:Access facility that allows a single SOLVE terminal to be used to provide full screen access to any number of other applications. The user may 'jump' from one application to another using designated 'jump' keys on the keyboard, or special command strings.

MAI-OC (MAI/Operator Control)

A SOLVE management services facility that allows an operator in OCS to have LU Type-1 sessions with many other applications from an OCS window. When used in conjunction with MSGPROC NCL procedures this can provide automated central monitoring and operation of multiple applications from the one operator console.

Management Services

This the central core of functions and service routines within SOLVE. It supports all of the SOLVE products.

Management Services Connection

A SOLVE management services connection is a communication link from SOLVE management services to the external database. It is started, stopped, defined, and deleted using the EDB command. A management services connection must be defined and started before opening an NCL connection. See also *NCL Connection*.

MDO (Mapped Data Object)

Any data item that can be represented as a continuous string a bytes in storage.

Message partition

A term associated with NPF that describes the group of network resources for which a user ID will receive unsolicited (PPO) VTAM messages.

Modify Interface

A means of communicating with an application program from the system console in OS/VS systems. The MODIFY command, abbreviated to F, avoids having an outstanding REPLY at the system console. SOLVE management services supports the use of the MODIFY command.

MSGPROC

An NCL procedure used to intercept and process messages destined for a user's Operator Console Services (OCS) window.

MSP

An operating system for large scale Fujitsu systems.

MVS (Multiple Virtual Storage)

IBM's major operating system. Fujitsu's functional equivalent is MSP.

NAU

See *Network Addressable Unit (NAU)*.

NCL (Network Control Language)

The interpretive language that allows logical procedures (programs) to be developed externally to SOLVE management services and then executed by SOLVE management services on command. NCL contains a wide range of logic, built-in functions and arithmetic facilities which can be used to provide powerful monitoring and automatic control functions.

NCL Connection

An NCL connection is a communication link from an NCL procedure to the external database. An NCL connection is opened and closed with the &EDB verb. A SOLVE management services connection must be defined and started before opening an NCL connection. See also *Management Services Connection*.

NCL Procedure

A member of the procedures dataset comprising NCL statements and SOLVE or VTAM commands. The NCL statements and other commands are executed from an EXEC or START command specifying the name of the procedure.

NCL Process

The NCL task that is invoked, usually by a START command to execute one or more associated procedures. Each NCL process has a unique NCL process identifier.

NCL Processing Environment

Provides the internal services and facilities required to execute NCL processes for the user, from its associated SOLVE window.

NCL Processing Region

All users (real or virtual) have an NCL Processing Region associated with their user ID while logged on. This region provides all of the internal services needed to allow the user to have processes executed on their behalf. There may be a maximum of two active NCL environments in a user's NCL region.

NCLID

A 6 digit NCL process identifier which is unique within the system. It is used to identify a process for the purpose of communicating with that process.

NCL/EF (NCL/Extended Function)

An extension to NCL which provides a relational database facility that can be used as a repository for applications running within a SOLVE system. Full update capabilities, including scans with extensive boolean logic, are provided.

NCP (Network Control Program)

This resides within and controls the operation of a communications controller. The NCP communicates with VTAM.

NCS (Network Control Services)

A facility of the SOLVE:Netmaster that provides full screen displays and navigation of the network in SOLVE management services.

NET/MAIL

A SOLVE product that provides a personal electronic mail facility for network users.

Network Addressable Unit (NAU)

In SNA, a logical unit, a physical unit, or a system services control point. The NAU is the origin or destination of information transmitted by the path control network. Synonymous with network accessible unit.

Network Control Language

See *NCL*.

NEWS (Network Error Warning System)

A facility of the SOLVE:Netmaster which is used to provide network error and traffic statistics and error alert messages.

NMINIT

The NCL procedure automatically executed after system initialisation has completed. It cannot contain commands that require VTAM facilities as it is executed before the primary ACB is opened. The procedure name can be changed by the installation.

NMREADY

The NCL procedure automatically executed once system initialisation has completed. It can contain commands that require VTAM facilities as it is executed after the primary ACB is opened. Procedure name can be changed by the installation.

Node

A connection point in a communications network.

Non-Roll Delete Message

See *NRD (Non-Roll Delete) Message*.

NPF (Network Partitioning Facility)

A facility of SOLVE management services that allows the range of resources which an operator can influence to be denied.

NPF Control Member

A member of the NPF dataset member which defines a list of member names that are to be the resource tables for the associated user ID.

NPF Resource Table

A member of the NPF dataset that defines a group of network resource names. The resource names can be defined specifically or generically using wildcard characters. A resource table is pointed to by a control member.

NRD (Non-Roll Delete) Message

A message that will not roll off an OCS window display until explicitly deleted. See *DOM*.

NT 2.1

Node Type 2.1. A node in an SNA network. It implements a peer-to-peer protocol and allows greater dynamics in network configuration, greater independence in session set up between partner LUs and reduced definitions.

NTS (Network Tracking System)

A facility of the SOLVE:Netmaster used to provide SNA session monitoring, dynamic online network tracing, accounting, and response time information in conjunction with diagrammatic representations of session partners.

OCS (Operator Console Services)

A facility of SOLVE management services that provides general operational control and an advanced operator interface to VTAM for network management.

OS/MVS (Operating System/Multiple Virtual Storage)

see *MVS*.

OSI (Open Systems Interconnection)

A set of ISO standards for communication between computer systems.

Packet

The logical unit of transmission in a network.

Panel Maintenance

A facility of SOLVE management services which allows users to generate and modify panel definitions used for presentation purposes by NCL procedures. In releases prior to Version 3.0, this function was known as Edit Services.

Panel Skip

The ability to chain menu selection requests together without having to display intermediate selection panels.

Password

A 1 to 8 character string chosen by a user and linked to their user ID for security purposes. To gain access to the system a user must enter both their defined user ID and its associated password.

PDS (Partitioned DataSet)

A type of dataset format that supports multiple individual members in the one physical dataset. Equivalent to the Source Statement Library in VSE/SP systems.

Peer-to-Peer Management

A non-hierarchical heterogeneous network management system.

PFK (Program Function Key)

See *Function Key*.

Physical Unit (PU)

The control unit or cluster controller of an SNA terminal. The part of a control unit or cluster controller which fulfils the role of an SNA-defined physical unit.

PIU (Path Information Unit)

An SNA packet.

PLU (Primary Logical Unit)

Relates to SNA. A type of LU that is usually used by the application programs in a host. It refers to the BIND sender for a session.

PPI (Program to Program Interface)

PPI is a general-purpose facility which allows programs, written in any language, to exchange data.

PPO (Primary Program Operator)

A VTAM term that describes a facility of VTAM that allows unsolicited network messages to be delivered to an application program, such as SOLVE management services, for processing.

PPOPROC

The name given to the NCL procedure used to intercept unsolicited VTAM (PPO) messages .

Preload

A term applied to NCL procedures which are loaded into the system before being required, to improve system performance .

Primary and Secondary

(SNA) Primary and secondary are SNA terms for describing the LU's role when the session is established. The primary LU sends the BIND request that causes the session to be established, and the secondary LU receives the BIND request. Rules defined in the BIND request determine which of these is the first speaker in the exchange of information.

Print Services Manager (PSM)

PSM is a facility of SOLVE management services which simplifies the control of the physical printing of reports on JES or network printers.

Program to Program Interface (PPI)

PPI is a general-purpose facility which allows programs, written in any language, to exchange data.

PSM

See *Print Services Manager (PSM)*.

PU

(SNA) Physical Unit. Each node (a logical grouping of hardware) in an SNA network is addressed by its PU. There are 4 types of nodes or PU in an SNA network: PU-T5, PU-T4, PU-T2, PU-T1. See *PU Type x*. A PU is a type of NAU. (See *Network Addressable Unit (NAU)*).

PU Type 1

(SNA) A type of Physical Unit or Node in an SNA network. Consists of a terminal (such as an IBM 3278).

PU Type 2

(SNA) A type of Physical Unit or Node in an SNA network. Consists of a cluster controller (such as an IBM3274, 3276, 3770 or 3790).

PU Type 4

(SNA) A type of Physical Unit or Node in an SNA network. Consists of a communications controller (such as an IBM 3704, 3705, 3725 or 3745).

PU Type 5

(SNA) A type of Physical Unit or Node in an SNA network. Consists of a host computer system (such as an IBM30xx, S/370 or 43xx, running VTAM or sometimes VCAM).

Remote Operator Facility (ROF)

A facility of SOLVE management services that allows an operator to sign on to a remote location, execute commands and have the results returned.

Remote Terminal Access Method (RTAM)

A facility that controls operations between remote terminals and job entry subsystems.

Reply

The information returned to a directive as a result of a request. This information may be either a response or an exception, together with appropriate arguments. See *Exception* and *Response*.

Request

The invocation of a directive, together with appropriate arguments. See *Exception* and *Response*.

Request Unit

(SNA) A message unit that contains control information such as a request code, or function management headers, end-user data, or both.

Reserved word

The term given to a token that will terminate an input expression if found unquoted outside the topmost parenthesis level. In the context of a particular verb statement the verb keywords are reserved words. A reserved word has special meaning for the current statement only, and different statements have different reserved words. Note that there are no words that are reserved throughout NCL.

Resource Table

A term associated with NPF that describes a list of resource names or generic resource names that define a command or message partition .

Response

The success result of a service request. See *Exception* and *Reply*.

Response Unit

(SNA) A message unit that acknowledges a request unit.

Return Code

A code returned from the system that indicates the success or failure of the task performed.

ROF

See *Remote Operator Facility (ROF)*.

RTAM

See *Remote Terminal Access Method (RTAM)*.

RTM (Response Time Monitor)

A facility provided by IBM's 3x74 control units to monitor end-user response times. NEWS can interpret this data .

RTM (Response Time Measurement)

Measurement of the time which passes between the user starting an action (by pressing a key) and the response appearing on the screen.

RU (Request/Response Unit)

(SNA) A generic term for a request or response unit.

SAW (Session Awareness Data)

A type of network management data supplied by VTAM and processed by NTS.

Security Initialisation Unit

A hardware device that creates and loads encrypting codes, also known as *keys*, for your computer system.

Sequence Number

A number assigned to each message exchanged between a VTAM application program and a logical unit. Values increase by one throughout the session, unless reset by the application program using an STSN or CLEAR command .

Server

A process designed to serve the data to a client, or request process, for one or more users.

Session Name

A name assigned to a workstation or session to permit it to receive messages or share resources.

SIS (Screen Image Services)

A part of the MAI/EF facility of SOLVE:Access which provides the ability to record screen images for later retrieval or to send to another user.

SLU (Secondary Logical Unit)

(SNA) A type of LU that is usually used by the end-users at the terminals or by programs which reside in the peripheral node.

SMF (System Management Facility)

An optional control feature of OS/VS that provides the means for gathering and recording information that can be used to evaluate system usage.

SNA (Systems Network Architecture)

This term describes the logical structure, formats, protocols, and operational sequences for transmitting communication data through the communication system (Fujitsu equivalent is FNA). A set of standards that allows the integration of all the different IBM hardware/software products into a universal network. Introduced in 1974.

SOLVE

This is the name of a suite of systems management products which assist in the management and operation of an organisation's information processing environment.

SOLVE:Netmaster

A SOLVE product which addresses the problems of Network Management. It comprises NCS, NTS and NEWS.

SRF (Session Replay Facility)

A part of the MAI/EF facility of SOLVE:Access which provides the ability to record and playback terminal session scenarios .

Structured field

Representation of user ID attribute information exchanged between SOLVE management services and its security exit.

Subtask

A unit of work whose environment is established by a main task, but has its own TCB, and is displaceable by the operating system.

SYSPARMS

System parameters—values that affect some SOLVE system capabilities. Some SYSPARMS can be modified dynamically.

Thread

A unit of work running under the control of an application program.

Timestamp

The instant of time at which the information described by a data item was valid.

TSO (Time Sharing Option)

Allows terminal operators to interact directly with computer resources and facilities. Used mainly by application and system programmers. (Fujitsu equivalent is TSS).

UAMS (Userid Access Maintenance Sub-system)

The security component of SOLVE management services that supports the definition of authorised users and their associated function and privilege levels.

UCS (Universal Character Set)

A printer feature that permits the use of a variety of character sets.

UDB (User Data Base)

1. UDB file access method layer allowing file access from NCL.
2. A term used to identify VSAM datasets to which NCL procedures may have access using the &FILE verb (GET, PUT, ADD, and DEL options).

UDM (UnDeliverable Message)

A term that applies to the Network Partitioning Facility (NPF) of SOLVE management services. It describes a message that cannot be directed to a terminal operator partitioned for the resource to which the message refers, or a message that does not apply to a specific resource.

User ID

Defines the function and privilege level to which a specific user is entitled when they sign on to the system. It is associated with a secret password to prevent use by unauthorised personnel. This definition is stored in the UAMS dataset or on an external security system.

USS (Unformatted System Services)

A VTAM term that describes a facility that translates an unformatted command such as LOGON or LOGOFF, into a field formatted command for processing by formatted system services. Applies to terminals before connection to an application.

Verb

The term given to a stand-alone statement in an NCL program. NCL *verbs* cause actions to occur. There are different types of verbs, some that dictate the flow of processing and logic, others that fetch information for the procedure to process and others that cause data to flow to external targets.

VFS (Virtual File Services)

The VSAM dataset, used by many facilities as a database.

VM (Virtual Machine)

A superset operating system that allows other operating systems to run as if they each had their own machine.

VM/ESA

An enhanced version of VM that supports 31-bit addressing.

VM/XA (Virtual Machine/Extended Architecture)

An extended version of VM.

VOS3

A Hitachi operating system comparable to IBM's MVS.

VR

(Session) Virtual Route.

VSAM (Virtual Storage Access Method)

A method for processing data files that utilises relative, sequential, and addressed access techniques. Conceptually identical to ENSCRIBE.

VSE (Virtual Storage Extended)

An operating system for small IBM systems.

VTAM (Virtual Telecommunications Access Method)

A suite of programs that control communication between terminals and application programs.

Wildcard

The term used to describe the character used (usually an asterisk) when defining resources generically — no specific matching character is required in the wildcard character position.

XSP

A Fujitsu operating system. Successor to FSP.

XNF

A Hitachi network access method for OSI networks.

Symbols

&FTS variables, 7-1, 7-9, 8-2, 8-11
\$@COPY, 9-8
\$@DELETE, 9-8
\$@L, 9-8
\$@M, 9-8
\$@NMFTS, 9-9, 9-10, 9-12, 9-15
\$@NOTIFY, 9-9
\$@R, 9-9
\$@REPRO, 9-9
\$@S, 9-9
\$USERBCI, 4-4, 4-7, 4-8, B-2, B-4
 customising, 4-9
\$USTSFTS, 8-2, 8-11, 8-15, 9-11

Numbers

3270
 datastream, 5-3
 full-screen conversation, 5-3
 interface, 1-2, 3-6, 4-7, 5-1, 5-3, 5-4
 See also Virtual 3270 Interface
 Model-4, 4-7

A

access authorisation exit, 7-6, 8-8
Access Control Block (ACB), 1-3, 2-2, 3-2,
 4-5
access to VFS, 6-4

activity log, 4-3
ad-hoc file transmission, 8-9, 8-16
autologon, 2-2, 2-4, 3-2, B-4, B-6

B

Batch Command Interface (BCI), 1-2, 4-3, 4-5,
 4-6, 4-7, 4-9, 4-11, 5-3, B-2
batch execution, 3-6
batch processing environment, 4-1
BCI command exit, 4-1, B-4
BCI control commands, 4-1, 4-9
 BCIDISC, 4-6, 4-8
 BCIEXITC, 4-6, 4-10, B-4
 BCILOGON, 4-4, 4-5, 4-7, 4-10, B-5
binary full word, 4-10

C

calling program, 5-2
calling sequences, 7-1
card-image format, 4-1
checkpoint information, 6-3
command authority check, 1-4
command data, 4-2
command exit, 4-3, 4-5, 4-9, 4-10
command processor, 2-2
conditional definition deletion, 7-7
control information, 4-7
control statements, 9-9

core-image library, 4-6, B-2
cross-editing functions, 8-2

D

dataset organisation, 9-4
dataset passwords, 7-10
dataset transmission, 7-3
DCB attributes, 8-6
debugging BCI NCL procedures, 4-9
definition
 copy, 8-15
 create, 8-3, 8-15
 delete, 8-8, 8-14
dialogue
 connecting to SOLVE system, 1-3
 continuing, 1-4
 FTS ISFP, 9-1
 messages, 9-8
 procedures, 9-3, 9-14, B-8
 programs, 9-4
directory block information, 8-6
directory blocks, 7-5
distributed procedures, 7-1
documentary messages, 4-7
DOS systems, 4-2, 4-3
dynamic transmission definition, 7-11

E

EASINET, 2-2, 2-3, 2-4, B-6
edit restrictions, 9-3
error recovery, 9-15
exiting, 2-4, B-6
express logoff, B-6
external interface, 8-11, 9-1
 components, 1-2, 1-4, B-3
 operational environment, 1-4
 requirements, 1-4
 tailoring, 6-2
 types, 1-2
 batch command, 1-2
 TSO command, 1-2
 TSO full-screen, 1-2
 virtual 3270, 1-2

F

field descriptor list, 5-2, 5-3, 5-4
file transfer, 9-1, 9-5, 9-12
 direct, 9-2
 staged, 9-2
File Transmission Services (FTS), 1-2, 2-3, 3-4,
 3-7, 6-1, 7-11
 See also FTS
 &FTS variables. *See* Transmission ...
 active transmission status, 7-8
 completion notification, 9-13
 definition maintenance, 6-3, 7-9
 failed transmissions, 7-11
 failure, 9-15
 initiator, 6-3
 limitations of procedures, 7-10
 NCL interface, 6-1, 6-2, 7-7, 8-1, 8-10,
 8-11, 9-1, B-5, B-7, B-11
 non-FTS sub-commands
 any other data, 3-5
 HELP, 3-4
 MAXRC, 3-4
 MORE, 3-4
 READ, 3-4
 procedures, 6-3
 status information, 7-8
 sub-command environment, 8-1
 sub-commands
 END, 3-4
 FTSDEF, 3-4, 8-8, 8-10, 8-11, 8-12,
 8-15
 FTSDEL, 3-4, 8-8, 8-12
 FTSSEND, 3-4, 8-10, 8-12, 8-15, 9-5
 FTSXMIT, 3-4, 8-10, 8-12, 8-15
 transmission class, 7-2, 7-8, 8-4
 transmission definition, 6-3, 6-4, 7-1, 7-3,
 7-7, 7-9, 7-12, 8-4, 8-8, 8-10
 See also Defining the transmission
 transmission direction, 7-8
 transmission request, 7-1, 7-6, 7-9, 7-11,
 8-7, 8-8, 8-13, 8-15, 9-3, 9-7, 9-14,
 9-16, B-9
 transmission request names, 9-14, 9-15
 transmission status, 6-3, 6-4, 7-8
 transmission supervision menus, 6-3
 TSO interface, 9-1
FROM command, 8-7, B-9
FTS. *See* File Transmission Services
full-screen dialogue, 5-1
full-screen panels, 6-3, 7-12, 9-12, B-5
function keys
 F10, 7-8

F11, 7–8
F7, 7–8
F8, 7–8

H

Help, 3–4
help panels, 9–7

I

image copy, 7–5, 8–6
input command line, 4–7
Inter–Management Services Connection (INMC), 6–3
interfaces
 common features, 1–3
 common requirements, 1–3
 SOLVE management services, 1–3
invalid call, 8–13
ISPF, 3–7, 9–12, 9–16
 dialogue, 9–8, 9–10, 9–14, B–7, B–9
 dialogue application, 9–1, 9–13
 dialogue components, 7–1, 7–11
 dialogue interface, 9–1, B–7, B–9, B–11
 dialogue manager, 1–2, 3–8, 6–2, 9–3, 9–5
 dialogue panels, B–7
 dialogue procedures, B–7
 options, 9–16
 overview, 9–1
 panel, 8–15
 profile dataset, 9–13
issuing commands
 SOLVE, 3–5
 SOLVE management services, 3–5
 VTAM, 3–5

J

JCL
 conditional statements, 4–1
 execution JCL, 7–2, B–10
 return codes, 4–1, 4–7, 4–11

L

link failure, 7–3, 9–14
load library, 4–6, B–1, B–8

local host, 9–2
logging off, 2–5, 4–6
logging on, 2–4
 user ID & password, 1–3

M

message handling, 9–12
messages returned to TSO, 8–13
MSP environment, 2–1, 3–1
MSP operating system, 1–2
multi-member transmission definition, 8–8
MVS operating system, 1–2

N

name masking, 7–11
NCL
 See also Network Control Language
 procedures, 1–4, 2–6, 3–1, 4–1, 4–7, 4–9,
 5–1, 5–3, 5–4, 6–2, 6–4, 7–1, 7–7, 7–10,
 7–12, 8–10, 9–3, 9–9, 9–11, 9–12, 9–15,
 B–5, B–8, B–11
 See also Network Control Language
 &INTREAD, 3–5
NMCMD, TSO commands, 1–2
NMFTSID, 9–3, 9–4, 9–10, 9–15, 9–17
NMLOGON, TSO commands, 1–2

O

operating systems
 MSP, 1–2
 MVS, 1–2
 VSE, 1–2
operational environment, 9–10
 NCL interface, 6–2
Operator Control Services (OCS), 6–3, B–5
OS systems, 4–2, 4–3, 4–6, 4–12, B–2, B–11
OS/MVS environment, 2–1, 3–1
outstanding responses, 3–4

P

panel characteristics, 5–3

panel definition, 5-3
Partitioned Dataset (PDS), 3-8, 7-11
partitioned dataset (PDS), 8-4, 8-5, 8-6, 8-16,
8-17, 9-3, 9-5, 9-8, 9-16, 9-17, B-10
pass-through-dialogue, 1-2
PDS transmission, B-10
positional parameters, 3-2
private definition, 7-2, 7-6, 7-7, 7-9, 7-12,
8-13, 8-15, B-9, B-11

R

remote host, 9-2
return codes, 3-6, 4-1, 4-10, 4-11, 8-14,
8-15, 9-5
JCL, 4-1, 4-7, 4-11
ROF session, 9-13

S

scrolling, 7-8
security exit, B-3
SOLVE management services
activity log, 1-4
aplname, 4-10
connecting to, 1-3, 3-2, 6-2
distribution library, 5-1
forms of access, 1-1
initial menu selection, 2-3
interfaces, 1-3
internal command environment, 3-5
issuing commands, 3-5
multiple user ID, 2-4
security, 1-3
signing on, 1-3, 2-4
standard facilities, 1-1
standard session, 1-4
system dataset, 6-4
user ID, 2-4, 3-2, 4-4
user services, 3-1, 3-5, 4-1, 4-4, 4-7, 4-9,
5-1, 5-3, 8-1
utilities, 6-2, B-8
VFS VSAM database, 6-3
space allocation
directory, 8-6
primary, 7-4, 8-6, 9-5
secondary, 7-4, 8-6, 9-5
type, 7-4

staging dataset, 9-2, 9-4, 9-6, 9-8, 9-12,
9-14, 9-15
standard assembler modules, 1-2
standard definition panel, 7-9, 7-10
status display filtering, 7-11
See also Display filtering
sub-commands, 3-1, 3-3, 3-4, 3-6, 3-8, 6-2,
8-1, 8-10, 8-11, 8-15, 8-17, B-7
SYSOPER, 7-12
SYSPARMS, 6-4, B-5, B-9
system access privileges, 1-4
system definition, 7-2, 7-12, 8-7, 8-15, B-9

T

tailoring, 6-1, 6-2, 7-9, 7-11, 9-4, 9-9, 9-10,
9-17
\$USERBCI, 4-9
temporary datasets, 9-10, 9-11, 9-13, 9-14
TO command, 8-7, B-9
TSO
See also CLIST
automatic logon, 2-4
CLIST, 3-3, 3-6, 3-8, 6-2, 8-17, B-6
command interface, 1-2, 3-1, 5-3, 6-2, 8-1,
8-11, 8-15, B-3, B-5, B-7
command processor
NMCMD, 1-2, 3-1, 3-2, 3-4, 3-5, 3-6,
3-7, 3-8, 6-2, 8-1, 8-3, 8-8, 8-11,
8-15, 8-17, 9-1, 9-3, 9-5, 9-6, 9-8,
9-16, B-1, B-2, B-4, B-7
NMLOGON, 1-2, 2-1, 2-2, 2-4, 2-6,
B-1, B-2, B-6
using, 8-15
command processors, 1-2, 2-2
dataset prefixing, 8-4
FTS interface, 7-1
full-screen interface, 1-2, 1-4, 2-1, 2-4,
2-5
See also Full-screen session
full-screen session, 2-1
full-screen interface, B-3, B-5, B-6
help, 3-4
PUTLINE macros, 8-2, 8-7, 8-9
return codes, 2-3
session, 2-2
sub-command processor, 3-4, 3-5
support for TSO users, 6-2
user ID, 2-2, 2-4, 9-15

U

- unit information, 8-4
- unrecoverable open error, 4-5
- unrecoverable transmission failures, 9-10
- User Access Maintenance Subsystem (UAMS),
 - 1-3, 4-4, B-3, B-4
- user command exit, 4-1
- user database (UDBs), 6-4, 7-12
- user services, 8-11, B-2, B-4, B-6
- user-defined sub-commands, 3-6
- using NMCMD, 8-15
- utility load module, 1-2

V

- VFS, 7-1, 7-8, 7-9
 - database, 6-3, 8-8, 8-11, 8-13
 - file error, 8-15
- volume-serial information, 8-4
- VSAM dataset, 9-3
- VSE operating system, 1-2, 4-6, 4-12, 6-4,
 - 7-2, B-2
- VTAM
 - access control block, 1-3, 2-2, 4-6, 5-2
 - APPL definitions, B-3
 - application to application session, 1-3
 - definitions, 1-3
 - domain, 2-1, 3-1, 4-1, 5-1
 - issuing commands, 3-5

