

CA SOLVE:FTS

User Guide

Release 12.1



Second Edition

This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the "Documentation") is for your informational purposes only and is subject to change or withdrawal by CA at any time.

This Documentation may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This Documentation is confidential and proprietary information of CA and may not be disclosed by you or used for any purpose other than as may be permitted in (i) a separate agreement between you and CA governing your use of the CA software to which the Documentation relates; or (ii) a separate confidentiality agreement between you and CA.

Notwithstanding the foregoing, if you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2010 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At <http://ca.com/support>, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at <http://ca.com/docs>.

Contents

Chapter 1: Introduction 11

How Data Set Transmission Works	11
Transmission Definitions	11
Transmission Requests	12
Transmission Initiators	13
Request Scheduling	14
Concepts Behind the Control of Transmissions	15
Error Recovery and Transmission Restarts	15
Transmission Status Information	15
Activity Reporting.....	16
Initiators and Network Routing.....	16
Initiators and Data Compression.....	17

Chapter 2: Getting Started 19

Access CA SOLVE:FTS.....	19
Authorized Users.....	20
Private and System Requests.....	20
Access Masks.....	21
Lists.....	21
Scrolling.....	22
Search for an Item.....	23
Online Help.....	23
Tip of the Day.....	23
Help About a Panel.....	24
Help for a Message	24
Work in Two Windows	25
Split Screens.....	25
Swap Screens	25

Chapter 3: Maintaining Transmission Definitions 27

Add Transmission Definition	28
List Transmission Definitions.....	29
Maintain Transmission Definitions.....	29
Operating System Dependencies	30

Chapter 4: Transmission Supervision **31**

Access FTS : Transmission Supervision Menu	31
Transmission Requests	31
Display Transmission Requests	32
Transmission Definitions	32
Display Transmission Definitions	32
Request Outbound Transmission	32
Transmission Activity	33
Display Transmission Activity	33
List Request Completion Codes	33

Chapter 5: Initiator Supervision **35**

Initiator Supervision	35
Access FTS : Initiator Supervision Menu	35
FTS : Initiator Supervision Menu	36
Display Active Initiators	36
FTS : Transmission Initiator Supervision Panel	37
List Destinations	42
Display List Definitions	43

Chapter 6: Using Operator Console Services **45**

Operator Console Services	45
Access OCS	45
OCS Panel	46
Command Line	46
Operating Mode Indicators	46
Roll Delete Area	46
Non-roll Delete Area	47
Roll-delimiter Line	47
Title Line	47
Time Display	47
Run Multiple OCS Panels	47
Set Window IDs	48
Function Keys	49
Conversational Function Keys	49
Prefix and Suffix Function Keys	49
Assign Your Own Values to Function Keys	50
Specify Commands to Function Keys	51
Specify Function Keys Using NCL Procedures	52
Use Commands in OCS	53

Command Authority Levels	53
Abbreviate Commands	53
Reuse Commands	54
Rename Commands	55
Monitor and Control in OCS	56
Control Message Presentation Speed	56
Unwrap Messages	57
Clear the OCS Window	58
Receive Non-roll Delete Messages	58
Hide NRD Messages	59
Restore Hidden NRD Messages	59
Delete NRD Messages	59
Use NRD Messages with ROF Sessions	60
Use the Activity Log to Help Monitor Your Regions	60
Record Additional Information in the Activity Log	60
Interpret Messages and Codes to Help Monitor Your Region	61
Issue Commands	61
Issue Commands in Background Environments	61
Issue Commands at Specified Times	62
Execute or Start NCL Processes from OCS	66
Execute NCL Processes Serially	66
Execute NCL Processes Concurrently	67
NCL Identifiers	67
Execute an NCL Process from a Serial or Concurrent Process	67

Chapter 7: Using Commands to Control Transmissions 69

Use of Commands	69
PROFILE Command	69
TRANSMIT Command	69
FTSMOD Command	70
FTSINIT Command	70
SHOW FTS Command	70
Timer Commands	71
Batch Jobs	71

Chapter 8: Using the TSO Interface 73

TSO Interface	73
Subcommand Environment	74
Get Help in the Subcommand Environment	75
NMCMD FTSDEF Subcommand	76
NMCMD FTSDEL Subcommand	82

NMCMD FTSEND Subcommand	83
NMCMD FTSXMIT Subcommand.....	84
TSO Interface NCL Procedures	84
Messages Returned to TSO	86
Return Codes from \$USTSFTS	89
Examples of Requests Using NMCMD	89
Example 1.....	90
Example 2.....	90
Example 3.....	91
Example 4.....	92

Chapter 9: Using the ISPF Dialog Interface **93**

ISPF Dialog Interface	93
ISPF Dialog.....	94
Direct File Transmissions.....	96
Staged File Transmissions	96
Edit Restrictions	96

Chapter 10: Using Print Services **97**

Print Services Manager	97
Access PSM.....	97
List Entries in the Print Queue.....	98
Display the Output of a Print Request	98
Modify a Printer Entry.....	99
Confirm Printing	99
Select the Printer	99

Appendix A: File Types and Attributes **101**

Supported File Attributes	102
Supplemental Data Set Organization	103

Appendix B: Generation Data Groups **105**

GDGs Support.....	105
GDS Support Example	106
Cold Start a Request Using GDGs	107
Potential Errors with GDGs	108
Collective Concatenation of Data Sets in a GDG	109

Chapter 1: Introduction

This section contains the following topics:

[How Data Set Transmission Works](#) (see page 11)

[Transmission Definitions](#) (see page 11)

[Transmission Requests](#) (see page 12)

[Transmission Initiators](#) (see page 13)

[Request Scheduling](#) (see page 14)

[Transmission Status Information](#) (see page 15)

[Activity Reporting](#) (see page 16)

How Data Set Transmission Works

CA SOLVE:FTS lets you move copies of (or members of) data sets from one location to another.

Transmitting a data set is a two step process:

1. The first step is *defining* the transmission. Transmission definitions are stored on a Virtual File System (VFS) VSAM database.
2. The second step is the actual file transmission request. The definition provided in the first step is *requested* through the Transmission Supervision Menu or the XMIT (or TRANSMIT) command from OCS. The transmission then proceeds when possible (when an initiator is available and the INMC link is active, and so on).

Transmission Definitions

All the information required to perform a particular transmission is stored on a VSAM database, the VFS, as a transmission definition. Each definition is identified by a unique 1- to 12-character name chosen when the definition is stored.

The existence of a transmission definition on the VFS does not itself cause a transmission operation to start.

Addition, modification, or deletion of transmission definitions is performed online by authorized users. All information is entered and displayed on full screen panels using the FTS : Transmission Definition Menu.

More information:

[List Transmission Definitions](#) (see page 29)

[Add Transmission Definition](#) (see page 28)

[Supported File Attributes](#) (see page 102)

[Supplemental Data Set Organization](#) (see page 103)

Transmission Requests

After a transmission definition is filed on the VFS, you can request CA SOLVE:FTS to execute that definition.

Execution of the definition causes the following actions:

- Information in the definition to be retrieved
- Appropriate data sets to be allocated
- Transmission to take place

The function of executing a definition is called a *transmission request*. To view the status and progress of the transmission request, use the FTS : Transmission Supervision Menu.

The transmission request takes the name of the transmission definition, and is identified by that name until the request has completed execution and, if required, has been acknowledged by the operator.

When CA SOLVE:FTS accepts a request, it takes a copy of the information in the transmission definition. This means that, if required, the transmission definition can be altered or deleted while the transmission request is still active or queued. However, if a request fails and the transmission definition is updated in an attempt to correct the error, the request must be deleted and reissued to pick up the changed definition.

You can issue a transmission request in *one* of the following ways:

- Select option T from the FTS : Transmission Supervision Menu.
- Select option L from the FTS : Transmission Definition Menu, and use the T (Transmit) action.
- Enter the **TRANSMIT** or **XMIT** commands from the Operator Console Services panel.

Transmission Initiators

The transmission of a request is handled by a *transmission initiator*. Each initiator processes one request at a time.

An accepted transmission request is placed on the FTS Request Queue pending the availability of a transmission initiator to handle its processing.

CA SOLVE:FTS maintains 16 initiators for each remote region with which it communicates, and as many of these initiators may be active as is required. There might be 16 initiators active to one destination but only two active to another, according to operational requirements.

Each initiator services one or more request classes. A request class is identified by a single letter, A through Z, assigned when the transmission definition is created. A transmission request can be processed only by an active initiator that services its particular class.

Initiators can be assigned either *normal* or *high* priority. A transmission request executing under a high priority initiator achieves a greater throughput rate than other requests executing under normal priority initiators at the same time. The priority of an initiator can be switched at any time and takes immediate effect if the initiator is currently executing a request.

The meaning of transmission classes is determined by the installation. For example, class A may represent high priority work and class E may represent low priority overnight report files. The allocation of classes to initiators enables your site to give preference to selected transmission classes or to stop transmission for particular classes.

You can temporarily change the allocation of initiator classes and the number of initiators active for a particular destination in one of the following ways:

- Using the FTS : Initiator Supervision Menu
- Using the FTSINIT command

The set of initiators for a destination is built the first time the link to that destination is started. The default set provides one active initiator servicing all classes, with the remaining 15 inactive and serving class A.

If this default set is not suitable, you can define an alternative initiator set for this destination from the FTS : Initiator Maintenance Menu. This definition is stored in the VFS database and is retrieved when the INMC link to the destination first becomes active. It can specify whether the CA SOLVE:FTS function is enabled for the destination on link activation. If the function is not enabled, operator intervention is required.

The functional split between initiator supervision and the definition of initiator sets enables temporary changes to be made to the initiators as operational requirements dictate (supervision) without making permanent changes to the defined initiator settings.

In addition to the control of the processing of outbound transmissions, the initiator functions enable the setting of a maximum inbound transmission limit for each remote region. This can be used to limit the number of transmissions that a region is prepared to accept, particularly if the need arises to reduce the amount of processing being performed.

Initiator definitions can also designate the type of compression performed on the transmitted data and may specify a preferred network route for the transmission if the INMC link was defined using the multipath options.

More information:

[Initiators and Network Routing](#) (see page 16)

Request Scheduling

When scheduling a request for processing by a transmission initiator, the following requirements should be satisfied:

- An INMC link to the remote region is specified as the transmission destination.
- Transmissions are enabled for the destination.
- A transmission initiator is available for the destination.
- The request must be of a request class serviced by that initiator.
- The inbound file limit has not been reached at the destination.

When these requirements are satisfied, the request is removed from the request queue and scheduled for processing by the appropriate initiator.

Concepts Behind the Control of Transmissions

The issuing of transmission requests and supervision of active requests is controlled from the region from which the requests are transmitted. The region that is receiving transmissions can only monitor the progress of inbound requests, acknowledge completed requests, and limit the number of concurrent transmissions that are to be received.

If a request fails for any reason, the restart of that request is always accomplished by the operator at the sending host. This means that even if the cause of the failure was rectified at the receiving host, the operator at the sending host must be informed that a restart should be attempted.

Error Recovery and Transmission Restarts

If a transmission request is scheduled but fails for a reason classified as unrecoverable, the request is requeued and suspended, pending clearance from an operator for it to be rescheduled. Such failures include data set errors, out-of-space conditions, invalid request parameters, operator interruption, and so on.

Requests that fail because of loss-of-contact conditions are rescheduled automatically when the link to the destination is restored, unless the AUTO-RESTART option is NO in the transmission definition. For most types of transmission, CA SOLVE:FTS checkpoints the progress of individual transmissions independently at both the transmitting and receiving regions, enabling a mutually acceptable restart point to be found when the request is rescheduled. If automatic restart is not permitted on a transmission, the request is placed in HELD status if it is interrupted by a loss-of-contact condition and is not restarted until released by operator intervention.

Note: Restart is not supported when the transmission is sent directly to spool (SYSOUT), or when either the input or output data set is a member of a partitioned data set.

Transmission Status Information

The status of file transmissions is available from the following sources:

- The FTS : Transmission Supervision Menu includes options to display active file transmissions and to list requests held on the database, including details of failed transmissions, completed transmissions, queued transmissions, and other status information. You can also invoke transmissions from this menu.
- The SHOW FTS command from OCS shows active file transmissions and the status of links to other regions.

Activity Reporting

Messages are logged and sent to OCS Mode screens with the appropriate profile (PROFILE FTSMMSG=YES,UNSOL=Y), to record start, stop, or restart of all transmission requests, including details of errors encountered (for failed requests) and statistical information (for successfully completed requests).

In addition, you can display requests that terminate because of errors through the Transmission Supervision function. This function identifies the cause of failure.

You must reset failed requests after the cause of failure is corrected and before the region attempts to reschedule them for restart. To reset failed requests, use the Transmission Supervision panels or the [FTSMOD command](#) (see page 70).

Note: Transmissions that are interrupted because of a link outage are restarted automatically on restoration of the link unless the automatic restart option has been suppressed.

A specific error code is associated with each failed request. For detailed explanations of the possible causes of these codes, select option CC from the FTS : Transmission Supervision Menu or enter the Q (Query) action against a request on the Transmission Supervision List.

Initiators and Network Routing

INMC links can take advantage of multiple physical routes through the network to maximize the use of available bandwidth and to minimize disruption of operations through the loss of one or more network paths. In some cases, however, it may be desirable to route certain bulk data traffic across one specific route—for example, certain high priority transmissions may be required to traverse a dedicated high speed link.

Each initiator can be assigned to a *preferred session* in an INMC session group. If this option is exercised, all transmissions of the initiator are directed across one specific session (provided the session is open).

Note: For more information about INMC link characteristics, see the *Reference Guide*.

Initiators and Data Compression

The data transmitted as part of a file transmission operation is compressed according to one of three supported compression algorithms:

- Bit-orientated data compaction
- Compression of strings of blank characters
- Compression of strings of any character

For each initiator, you can select the type of compression or suppress compression completely.

By default, bit-orientated data compaction is used.

Chapter 2: Getting Started

This section contains the following topics:

[Access CA SOLVE:FTS](#) (see page 19)

[Authorized Users](#) (see page 20)

[Lists](#) (see page 21)

[Online Help](#) (see page 23)

[Work in Two Windows](#) (see page 25)

Access CA SOLVE:FTS

To access the product, enter **/FTS** from the NetMaster : Primary Menu.

The FTS : Primary Menu appears.

Note: In addition to the FTS : Primary Menu, some functions are available to authorized OCS operators from an OCS Mode display, from the Command Entry panel, from the system console, or from NCL procedures.

More information:

[TRANSMIT Command](#) (see page 69)

[FTSMOD Command](#) (see page 70)

[FTSINIT Command](#) (see page 70)

[SHOW FTS Command](#) (see page 70)

Authorized Users

The access that a user has to functions depends on the privileges allocated in the definition of their user ID.

Note: For more information about user ID definition requirements, see the *Installation Guide*.

The functions fall into the following categories:

- Creating transmission definitions
These privileges classify a user as being allowed to define private definitions or system definitions, or neither.
- Issuing transmission requests
These privileges classify a user as being allowed to issue private requests or system requests, or both, or neither.
- Controlling CA SOLVE:FTS activity
In this category, the user may be entitled to exercise control over private or system requests and may be allowed control over the CA SOLVE:FTS environment. This includes starting and stopping initiators, changing classes and so forth.

Note: Within all categories, the requests or definitions that a user can reference are restricted to those with names that are valid for the user's private or system access and the associated access masks.

Private and System Requests

The concept of *private* and *system* requests let users carry out their own data set transmission requirements without being able to interfere with the transmission activities associated with the installation's production operations.

In general, individual users who need the data set transmission for private purposes are assigned privileges that enable them to define transmission requests for the movement of their own data sets, to request them, and to monitor the progress of those requests. They are not able to define or examine system definitions or request their execution.

Operational staff responsible for monitoring CA SOLVE:FTS activity are typically allowed control over the requesting of system definitions, and control over their progress and the overall CA SOLVE:FTS environment. These personnel may not necessarily be able to act as private users themselves and probably are not allowed access to system definitions.

Access Masks

An access mask is a 1- to 12-character string with which the name of a selected definition is compared. Access to the definition is granted only if the characters in the mask match the characters in the definition name.

Each user can be assigned two such access masks; one that controls their access to private definitions (if authorized) and one that controls their access to system definitions (if authorized).

Note: Masks are specified in the user ID definition.

Note: For more information about the implications of private and system users, and masks, see the *Administration Guide*.

Lists

Lists comprise a series of items that you can select or against which you can perform actions. The panel shows the actions that you can perform on the listed items.

There are four types of list:

Action Lists

Let you apply *actions* to one or more listed items. Enter the required action code beside the appropriate records.

Single Selection Lists

Let you select one item from a list (for example, the list of valid values for a data entry field) by doing *one* of the following:

- Entering the **S** (Select) action code beside the item
- Moving the cursor to a position anywhere in the line containing the item you want to select and pressing Enter

Multiple Selections Lists

Let you select one or more items in a list (for example, the list of panels used to customize your user profile).

Numbered Lists

Let you select a single item from the list by entering the appropriate number at the prompt (for example, the list of valid values for a data entry field).

Scrolling

If the listed information cannot fit onto the screen, use scrolling to access the off-screen information. You can scroll vertically and horizontally.

Scroll Vertically

Use the F8 (Forward) or F7 (Backward) function key to scroll the displayed information by the amount displayed at the Scroll prompt.

The following table shows valid scroll amounts.

Scroll Amount	Action
C (or CSR)	If scrolling forward, the line on which the cursor is currently positioned is moved to the top of the screen. If scrolling backward, the line on which the cursor is currently positioned is moved to the bottom of the screen.
D (or DATA)	The display is scrolled one full page, less one row, in the specified direction. If scrolling forward, the last line of the current page appears as the first line on the next page. If scrolling backward, the first line on the current page appears as the last line on the next page.
H (or HALF)	The display is moved half a page in the specified direction.
M (or MAX)	The display is moved to the beginning or the end of the displayed information, depending on the function key (Forward or Backward) used.
P (or PAGE)	The display is moved one full page in the specified direction.
<i>n</i>	The display is moved <i>n</i> lines in the specified direction. If you enter a temporary scroll amount at the command prompt (for example, Command ==> 5), then when you press the F7 (Backward) or F8 (Forward) function key, the displayed information is scrolled by the specified value once only.

Scroll Horizontally

Use the F11 (Right) or F10 (Left) function key to scroll the displayed information to the right or to the left.

Search for an Item

You can search for specific items in the retrieved information by using the F5 (Find) function key or the LOCATE command.

F5 (Find) Function Key

The F5 (Find) function key lets you find text in the retrieved information. Enter the text you want to find, and press F5. If the text contains more than one word, enclose the text in quotation marks.

You can press F5 again to find the next instance of the text.

For some lists, you can enhance the Find function in the following ways:

- Expand the search beyond the columns currently displayed by using the FMODE command
- Change the number of records searched between prompts by using the F PROMPT command

Note: For information about the FMODE and F PROMPT commands, see the online help.

LOCATE Command

The LOCATE command enables you to locate a particular record in a list. Enter **LOCATE** or **L** followed by a text string mask. The command locates the first record name that matches the mask.

Online Help

Online help is provided for panels and messages.

Context-sensitive help is available at different levels. When you are viewing a help panel, pressing F1 (Help) takes you to the next level of help available. Pressing F3 (Exit) takes you back to the previous level, or exits from help and returns you to the application. Pressing F4 (Return) exits help and returns you to the application immediately.

Tip of the Day

The region displays a tip about using the product at the bottom of the primary menu.

To display the detailed tip, place the cursor on the tip and press F1 (Help).

Help About a Panel

Panel-based online help includes information about what each panel is used for, how to complete the fields, the actions you can perform, and the use of available function keys. Use this online help to supplement the information in this guide while you are working in the region.

Press F1 (Help) to retrieve the online help for a given panel. When you are viewing a help panel, you can press F6 (HelpHelp) to find out how to use the help facility.

If the block of help text you require splits across two panels, use the arrow keys to move the cursor to the top or the bottom of the block and press F8 (Forward) or F7 (Backward) to bring the block into view.

Help for a Message

While you are working in the region, you receive messages that advise you of various events. These messages might be providing information only (for example, informing you that an update was successful). They might also alert you to errors (for example, if you try to enter an action that is not valid for a resource).

Each message has detailed online help text associated with it. Access the help text for a particular message in *one* of the following ways:

- If you are at a panel and a message appears in red on the third line of that panel, move the cursor to that line and press F1 (Help).
- If you receive a message referring you to the activity log for more detail, enter **/LOG** at the prompt to display the activity log.
- If you are using the activity log, a Command Entry panel, or Operator Console Services (OCS), you can do *one* of the following:
 - Move the cursor to the line displaying the message, and press F1 (Help).
 - Type the message ID at the prompt, and press F1 (Help).
- If you are viewing a transient log, enter **H** beside the message.
- You can also enter **/CODES** to display the Messages and Codes Menu that enables you to obtain help on messages and on miscellaneous error codes.

Work in Two Windows

You can divide your physical screen into two logical windows. Each window operates independently of the other, enabling you to perform multiple functions concurrently.

To open a second window in the region, press the F2 (Split) or F9 (Swap) function key.

When one window takes up the entire screen, the other window is considered *closed*.

Split Screens

Using the SPLIT command, you can perform the following actions:

- Split your screen horizontally and have one window above the other. Move the cursor to a row where you want to split screens, and press F2 (Split).
- Split your screen vertically and have two windows side by side. Move the cursor to any column on the bottom row, and press F2 (Split).
- Return a split screen to single window display in one of the following ways:
 - Move the cursor to the first line on your screen, and press F2 (Split) to minimize the window. The window containing the cursor disappears, and the other window expands to full size.
 - Enter =X to exit one of the windows. Your session with that window ends.

Swap Screens

Using the SWAP command, you can perform the following actions:

- Reverse the dimensions of the active window if you have two windows open and both are visible on the screen, and switch between them.
- Open a second full-screen window if you are currently operating with a single window open, and then switch between them.

To swap two full-screen windows

1. Display one of the panels.
2. Press F9 (Swap).
The primary menu appears.
3. Proceed to the second panel you require and press F9 (Swap).
The first of your swap-panels appears.
4. Press F9 (Swap) to switch between the two swap-panels.

Chapter 3: Maintaining Transmission Definitions

This section contains the following topics:

[Add Transmission Definition](#) (see page 28)

[List Transmission Definitions](#) (see page 29)

[Maintain Transmission Definitions](#) (see page 29)

[Operating System Dependencies](#) (see page 30)

Add Transmission Definition

To add a transmission definition

1. Enter **/FTSTD** at the command prompt.
The FTS : Transmission Definition Menu appears.
2. Select option A.
The FTS : Transmission Definition "FromFile" Details panel appears.

3. Complete the following fields:

Definition Name

Defines a name for the definition.

Limits: One through twelve characters

Destination

Defines the INMC link name of the remote region to which the data set is transmitted.

Data Set Name

(Optional) Specifies the name of the data set on this system to transmit.

DD Name

(Optional) Specifies the ddname of a data set that has been allocated.

Note: You must enter a Data Set Name or DD Name.

4. (Optional) Press F8 (Forward).
The FTS : Transmission Definition "ToFile" Details panel appears.
Review the settings, and update as required.
5. Press F3 (Exit).
The definition is saved.

Note: Only authorized users can access the Transmission Definition function. Such users can modify either *private* definitions or *system* definitions.

List Transmission Definitions

To list active transmission definitions

1. Enter **/FTSTD** at the command prompt.

Note: To filter the list, you can enter selection criteria, such as definition name or destination name.

The FTS : Transmission Definition Menu appears.

2. Select option L.

The FTS : Transmission Definition List appears.

From the list, you can browse, copy, delete, transmit, update, or add a definition.

3. (Optional) Press F11 to scroll right and display three additional pages:

- Page 2 displays FROM data set details.
- Page 3 displays TO data set details.
- Page 4 displays definition maintenance details.

Note: For more information about the actions and fields on these panels, press F1 (Help).

Maintain Transmission Definitions

When maintaining a transmission definition, you access the following panels:

FTS : Transmission "FromFile" Details

Displays details about the source data set for a transmission definition.

FTS : Transmission "ToFile" Details

Displays details about the target data set for a transmission definition.

When you select a definition to maintain, the first panel displayed is the "FromFile" Details panel. From there you can press F8 (Forward) to move forward to the "ToFile" Details panel.

Notes:

- There are two panels for the target data set; one describes a specific data set; the other describes SYSOUT details. Press F10 to select the relevant panel.
- For information about the fields and actions on these panels, press F1 (Help).

Operating System Dependencies

CA SOLVE:FTS presents as consistent an interface to the user as possible, regardless of the operating system under which it is executing.

Because file transmission may involve the movement of files between different operating systems, the information required when a transmission definition is being created or modified varies slightly according the system you are using.

The system type noted against the FROM panels indicates the panel that is presented when logged onto that type of system.

The system type noted against the TO panels indicates the panel that is displayed when transmitting to that type of system. Therefore, a user creating a transmission definition from CA SOLVE:FTS on a z/OS system, which is to transmit a file to CA SOLVE:FTS executing on a VM system, is presented with a z/OS FROM data set definition panel and a VM TO data set panel.

Chapter 4: Transmission Supervision

This section contains the following topics:

[Access FTS : Transmission Supervision Menu](#) (see page 31)

[Transmission Requests](#) (see page 31)

[Transmission Definitions](#) (see page 32)

[Request Outbound Transmission](#) (see page 32)

[Transmission Activity](#) (see page 33)

[List Request Completion Codes](#) (see page 33)

Access FTS : Transmission Supervision Menu

You can control the Transmission Supervision functions from the FTS : Transmission Supervision Menu.

To access the FTS : Transmission Supervision Menu

1. Enter **/FTSTS** from the command prompt.

The FTS : Transmission Supervision Menu appears.

Notes:

- For information about the options and fields on the menu, press F1 (Help).
- This option is only available to users who are authorized for control functions. Also, limitations may be imposed by a user's access mask.

Transmission Requests

The transmission request list provides an overall view of the file transmission activity in the region, for all destinations and for all requests, regardless of status. It includes active transmissions and queued transmission requests. You can specify additional search arguments to restrict the list to specific destination or origin, class, name, or status. Because wildcard processing is the default in this field, you need only specify the leading characters of the destination or request name.

The list is displayed in destination alphabetical order, with requests displayed in chronological order in destination. Outbound requests are listed first, followed by inbound requests.

Display Transmission Requests

To display transmission requests

1. Select option L from the FTS: Transmission Supervision Menu.

The FTS : Transmission Request List appears.

Transmission Definitions

The transmission definition list displays transmission definitions that meet the specified selection criteria. From the Transmission Definition List, you can browse the definition or request outbound transmissions.

Display Transmission Definitions

To display the transmission definitions

1. Select option LD from the FTS: Transmission Supervision Menu.

The FTS : Transmission Definition List appears.

Request Outbound Transmission

To request outbound transmission

1. Select option T from the FTS: Transmission Supervision Menu, and specify the Definition Name.

Either the transmission request is accepted or an error message is issued.

Note: If the link to the region to which the transmission is being made is not active, the request is queued and scheduled when the link becomes available.

When errors are displayed for failed requests or completion statistics are displayed for completed requests, they are prefixed with a six-character field that identifies the error and the location of the region that detected the error condition.

To review the error details for a failed request, enter **/FTSTS.L** and enter **Q** (Query) against the failed transmission request.

Transmission Activity

The transmission activity panel displays transmissions that are currently being serviced. The number of active transmissions displayed for any destination or origin is determined by the transmission limit that is defined in the initiator set definition.

Display Transmission Activity

To display transmission activity

1. Select option LA from the FTS: Transmission Supervision Menu.
The FTS : View Transmission Activity panel appears.

List Request Completion Codes

To list request completion codes

1. Select option CC from the FTS: Transmission Supervision Menu.
The FTS : Request Completion Codes panel appears.
2. Select a topic.
The six-character codes and summary explanations that FTS generates to acknowledge an error or a successful transmission appears. This information is in the format REM-*nn* or LOC-*nn*.

Chapter 5: Initiator Supervision

This section contains the following topics:

[Initiator Supervision](#) (see page 35)

[Access FTS : Initiator Supervision Menu](#) (see page 35)

[Display Active Initiators](#) (see page 36)

[List Destinations](#) (see page 42)

Initiator Supervision

Initiator Supervision provides operational control of transmission configuration. The relevant user ID must have control privilege to use these facilities. These facilities are available through the FTS : Initiator Supervision Menu.

Initiator Supervision lets you make temporary changes to the *active* set of initiators for a specific destination. It does not make permanent changes to the base initiator set definition for that destination.

Note: Modifications to initiator set definitions are performed through the FTS : Initiator Maintenance Menu. For more information, see the *Administration Guide*.

Access FTS : Initiator Supervision Menu

To access FTS : Initiator Supervision Menu

1. Enter **/FTSIS** at the command prompt.

The FTS : Initiator Supervision Menu appears.

FTS : Initiator Supervision Menu

```
----- FTS : Initiator Supervision Menu -----  
Select Option ==>  
  
    1  - Display, Modify Initiator(s)  
    2  - List Destinations  
    X  - Exit  
  
Destination Name ==>
```

The Initiator Supervision Menu contains the following options:

1 - Display, Modify Initiator(s)

Displays and modifies active initiator sets.

2 - List Destinations

Displays transmission destinations.

This menu contains the following field:

Destination Name

Specifies the name of the destination for which you want to display details.

Display Active Initiators

To display active initiators

1. Select option 1 from the FTS : Initiator Supervision Menu and enter the Destination Name.

The FTS : Transmission Initiator Supervision panel appears.

FTS : Transmission Initiator Supervision Panel

This panel shows the current initiator set used to service transmission requests to the region whose link name is specified as the destination name. The status of the link to that destination also appears. You can exit the panel by pressing F3.

----- FTS : Transmission Initiator Supervision -----						
DESTINATION PROD19		STATUS	IN-CONTACT		* IDLE *	
FTS ENABLED (Y/N) ==> Y			TRANSMIT LIMIT 16 (Set by receiving end)			
Init	Service Classes (A to Z)	Started	Com	Hi-Pr	Current Status	P/S
1	ABCDEFGHIJKLMN	Y	3	N	* IDLE *	-
2	A	N	3	N		-
3	A	N	3	N		-
4	A	N	3	N		-
5	A	N	3	N		-
6	A	N	3	N		-
7	A	N	3	N		-
8	A	N	3	N		-
9	A	N	3	N		-
10	A	N	3	N		-
11	A	N	3	N		-
12	A	N	3	N		-
13	A	N	3	N		-
14	A	N	3	N		-
15	A	N	3	N		-
16	A	N	3	N		-
MAXIMUM CONCURRENT INBOUND FILES (0 to 16) ==> 16					REFRESH (Y/N) ==> N	

Note: If the link to the selected region is not activated, no initiator set is built, and the request is rejected.

This panel contains the following fields:

DESTINATION

Displays the link name of the remote region.

STATUS

Displays the current overall status of the link to the remote region:

PENDING-CONTACT

Indicates that the INMC link was started but contact is not established.

IN-CONTACT

Indicates that the INMC link is active and communications are available.

LOST-CONTACT

Indicates that the INMC link is broken.

NOT-COMPATIBLE

Indicates that the remote region is not configured with a compatible version of CA SOLVE:FTS or is not configured with CA SOLVE:FTS.

A secondary status also appears to the right of the overall link status. It indicates the transmission status.

* ACTIVE *

Indicates that transmissions are in progress.

* DRAINING *

Indicates that the FTS ENABLED field was set to N. Active transmissions proceed to completion, but no new transmissions are started or accepted from the remote region.

* IDLE *

Indicates that transmission is enabled, but no traffic is active or available to schedule against the available initiators.

TRANSMIT LIMIT

Displays the maximum number of concurrent transmissions that can occur to this destination, regardless of the number of active initiators.

Note: This field cannot be changed. It reflects the MAXIMUM CONCURRENT INBOUND FILES value set by the remote region.

FTS ENABLED

Specifies whether transmission is enabled:

Y

Indicates that transmission is active for this destination. Transmission requests are scheduled to this destination if any are available, and inbound transmissions are accepted.

N

Indicates that transmission is not active for this destination. No new requests are scheduled, and no inbound transmissions from the remote region are permitted.

Setting this field to N lets currently active transmission requests, both inbound and outbound, continue but no more are started. While there are existing transmissions, the status **DRAINING** appears to the right of this field.

Init

Lists the initiator numbers. CA SOLVE:FTS maintains a total of 16 initiators for each destination. The set of 16 is referred to as the *initiator set*.

Service Classes

Each initiator can service between one and 26 request classes (A to Z). Duplicate classes are not allowed. The order in which the classes are specified in this field, left to right, determines the order in which the initiator is allocated work. So, if a single initiator is defined, servicing classes ADCB, and three transmission requests for classes A, B, and C exist, they are executed in the order A C B.

Each of the 16 initiators defined as STARTED must be assigned at least one class.

Example: The following panel shows an initiator set with five initiators defined, servicing twelve different classes among them. When multiple initiators are defined and a request is available for execution with a transmission class that is serviced by more than one (available) initiator, the request is scheduled under the initiator that has the highest preference for that class.

For example, if none of the five initiators shown in the following panel are busy and a class A transmission request becomes available for execution, CA SOLVE:FTS would find that:

1. Initiators 2 and 3 both service class A, but that initiator 2 accepts class A requests only if no class C or F requests are available.
2. Initiator 3 services class A in preference to any other class, and the request is assigned to initiator 3 for execution.

3. If a second class A request is issued while initiator 3 is still executing the first request, the second request would be assigned to initiator 2.

----- FTS : Transmission Initiator Supervision -----							
DESTINATION NEWYORK2		STATUS		PENDING-CONTACT		* IDLE *	
FTS ENABLED (Y/N) ==> Y				TRANSMIT LIMIT 16 (Set by receiving end)			
Init	Service Classes (A to Z)	Started	Com	Hi-Pr	Current Status	P/S	
1	XYZ	Y	3	Y	* IDLE *	2	
2	CFAB	Y	3	N	* IDLE *	-	
3	AB	Y	3	N	* IDLE *	-	
4	D	Y	3	N	* IDLE *	-	
5	PQRST	Y	3	N	* IDLE *	-	
6	A	N	3	N		-	
7	A	N	3	N		-	
8	A	N	3	N		-	
9	A	N	3	N		-	
10	A	N	3	N		-	
11	A	N	3	N		-	
12	A	N	3	N		-	
13	A	N	3	N		-	
14	A	N	3	N		-	
15	A	N	3	N		-	
16	A	N	3	N		-	
MAXIMUM CONCURRENT INBOUND FILES (0 to 16) ==> 16				REFRESH (Y/N) ==> N			

Started

Y indicates that the initiator is available for processing requests of its assigned classes. N indicates that the initiator is not available for processing requests.

Note: If this field is changed to N it prevents any further activity by that initiator. If the initiator is processing a request at the time, that request continues transmitting but no new requests are started when the current request ends.

Com

A value of 1, 2, 3, or N indicates the *compression level* to apply to transmissions performed by this initiator, where:

1

Indicates that blank string compression is required

2

Indicates that like string compression is performed.

3

Indicates bit level compaction. This is the default.

N

Suppresses compression on transmissions handled by the initiator.

Hi-pr

Y indicates that the initiator is allocated a higher transmission priority than normal priority (N) initiators. Transmissions executed under high priority initiators get preferential processing. But, the transmission of multiple concurrent requests to a particular destination causes all those requests to run more slowly than would be the case if only a single transmission was running.

Note: High priority should be used selectively—there is no advantage to making all initiators high priority.

CA SOLVE:FTS does not consider an initiator's priority setting when scheduling requests to initiators for execution. This means that if an initiator is always defined as high-priority so that requests of specific classes are always executed at high priority, the classes serviced by this initiator should not also be serviced by any other initiator; otherwise, there is no guarantee that requests intended to run at high priority are executed by the high-priority initiator.

In the example definition shown in the previous figure, initiator 1 is defined as high-priority and it services classes X, Y, and Z, which are unique to that initiator. This guarantees that class X, Y, and Z requests always execute at high priority.

Current Status

Specifies the current status. Valid values are:

IDLE

Indicates that the initiator is not processing any request, but is active and available for work of the nominated class.

(c) request-name

Indicates that the initiator is processing the transmission request, *request-name*, which is class c.

D--request-name

Indicates that the initiator is processing the transmission *request name* but has had its Started field set to N. The current request is continued, but no new request is started by that initiator. D indicates that the initiator is draining.

Note: To reverse the draining status, enter Y in the Started field.

P/S

If you want work to be split across available paths according to the INMC link definition as a whole, the field should be blank or contain a hyphen. If the designated session is not available, the work is serviced according to the INMC definition. The INMC link must have been defined with a transmission mode of preferential or rotate and backup. The value corresponds to the session number in the INMC link definition.

If you want to tie this initiator's work to a particular INMC session number, enter the number of the session.

Maximum Concurrent Inbound Files

Indicates the number of concurrent transmissions that this region is prepared to accept from the remote region. This value is reflected in the TRANSMIT LIMIT field at the remote region.

Setting this field to zero prevents the remote region from transmitting any data sets.

Setting this field to 16 permits the remote region to start up to 16 concurrent transmissions.

Setting the field to any intermediate value allows control over the inbound transmission count according to operational requirements.

Refresh

Specifies when the display is refreshed:

N

Updates the initiator set with any modifications when the Enter key is pressed and returns to the Initiator Supervision Menu.

Y

Refreshes the display each time the Enter key is pressed. This option enables continued displays of the initiator set status, for example, while waiting for all initiators to drain before terminating the link.

Any changes made to the panel are processed and the Initiator Set updated when the Enter key is next pressed.

List Destinations

List Destinations presents a panel that contains one line of information for each destination that has an initiator set built.

The information includes:

- Whether the destination is currently active
- The maximum number of concurrent inbound transmissions that can be received
- The user ID of the user that last changed the initiator set
- The time the change was made

Display List Definitions

To display list definitions:

1. Select option 2 from the FTS : Initiator Supervision Menu.
The FTS : Destination Status Information panel appears.

Chapter 6: Using Operator Console Services

This section contains the following topics:

[Operator Console Services](#) (see page 45)
[Access OCS](#) (see page 45)
[OCS Panel](#) (see page 46)
[Run Multiple OCS Panels](#) (see page 47)
[Function Keys](#) (see page 49)
[Assign Your Own Values to Function Keys](#) (see page 50)
[Use Commands in OCS](#) (see page 53)
[Monitor and Control in OCS](#) (see page 56)
[Receive Non-roll Delete Messages](#) (see page 58)
[Use the Activity Log to Help Monitor Your Regions](#) (see page 60)
[Record Additional Information in the Activity Log](#) (see page 60)
[Interpret Messages and Codes to Help Monitor Your Region](#) (see page 61)
[Issue Commands](#) (see page 61)
[Execute or Start NCL Processes from OCS](#) (see page 66)

Operator Console Services

Operator Console Services (OCS) lets you enter commands to control and monitor your resources.

OCS uses a formatted display panel called an OCS window to provide an environment for executing commands or NCL procedures. Your command results are returned to the window, with other system information, to provide a console function.

The level of authority granted in your user ID definition limits the actions you can perform in OCS. You control the way your OCS window looks and the way you use it. This set of attributes, privileges, and options is called your *operator profile*.

Certain attributes of your operator profile are controlled by UAMS. Other attributes can be changed using the PROFILE command.

Access OCS

To access OCS

1. Enter **O** at the prompt at the main Primary Menu.
The OCS window appears.

OCS Panel

The OCS panel has two distinct activity areas: a one-line command input area at the bottom of the window, and an output message display area called the roll delete area, which occupies the remaining space above the command line.

Command Line

The command line is the bottom line of the OCS window. The command line is the only display field where input is permitted. The cursor is automatically positioned to the right of the command line prompt when the panel is first displayed. To enter a command, position the cursor in the command line and press Enter.

Operating Mode Indicators

A mode indicator may appear to the left of your command entry area to indicate how the OCS window is currently operating. Values of the operating mode indicators and their meanings are as follows:

M (Monitor)

Terminal has monitor status and receives monitor messages.

P (Paused)

An NCL procedure has paused awaiting the entry of a GO, END, or FLUSH command. The SHOW NCL command gives you further details.

W (Waiting)

An NCL procedure is waiting for specific text to arrive. The SHOW NCL command gives you further details.

Roll Delete Area

When you receive messages as the result of commands entered on the command line, they are reported in the roll delete area with any unsolicited information you are profiled to receive.

Output to the roll delete area is written line-by-line from top to bottom of the screen. When the display area is full, new output messages wrap back to the top of the screen, overwriting the oldest displayed messages first.

Non-roll Delete Area

Any messages requiring a reply are delivered as non-roll delete messages. This means that the messages stay on your screen until you respond. These messages are displayed at the top of an OCS window above the roll delete area. The non-roll delete area is created only when a non-roll delete message is delivered to your OCS window.

Roll-delimiter Line

Messages are written from top to bottom of the screen. The next line for use is filled by a line of underscore (`_`) characters. This line is called the roll-delimiter line. It separates the oldest and newest output displayed. Output below this line is the oldest display information; output above the line is the most recent.

Note: The underscore characters used for this line can be changed using the PROFILE DELCHAR command.

Title Line

The top of the roll delete display area is reserved for a title that can be set or reset by using the TITLE command.

Time Display

The top left of the title line includes the present system time in *hh.mm* format and is automatically updated each time anything is written to the OCS window.

Run Multiple OCS Panels

You can use the [screen-splitting function](#) (see page 25) to run two OCS windows in parallel on the same screen.

You can have one screen window operating in OCS with the other part-screen or window in full-screen mode (for example, as a menu).

Set Window IDs

When two OCS windows are running simultaneously, it helps if you can distinguish each window while executing NCL procedures.

To set a name for each OCS window, enter the OCSID command followed by a 1- to 8-character name at the command prompt on an OCS window and press Enter. The name for each window appears to the right of the line, immediately above the command line.

Example: Set Window ID

To set a window ID of NET01, enter the following command:

```
OCSID NET01
```

After pressing Enter, NET01 appears to the right of the line immediately above the command line.

Note: You can set profile attributes for each OCS window, so that two OCS windows on the same terminal can have different profiles and appearances.

Function Keys

OCS windows have full support for 24 function keys. You can set function keys for each OCS window to suit your requirements. If you are running two OCS windows, each window can have a separate set of function key definitions.

A variety of function keys are available:

- Default (system-wide) function keys
- Immediate function keys
- Conversational function keys
- Prefix function keys
- Suffix function keys
- NCL controlled function keys

The OCS function keys are set to system default values when you enter OCS.

When you redefine a function key, its defined value applies only to the function key settings for your current window, and remains in effect only while your current window is active.

When you press an immediate function key, its assigned value is immediately entered into the system, without the need for further action by you.

Note: To discover the assignment of each function key, use the PF LIST command.

Conversational Function Keys

A conversational function key lets you modify its action before you release it for processing. When you press a conversational function key, its assigned value is displayed in the command line, so you can add to or modify the text. Press Enter to run the command after you modify it.

Prefix and Suffix Function Keys

A prefix function key assigns a set value as a prefix to the line of text where the cursor is located when that function key is pressed (that is, the command line or any other line in the OCS window display area).

Example: Prefix and Suffix Function Keys

The F6 function key is defined using the following command:

```
PF6 PREF,MSG USER1+
```

When you enter a message in the command line saying: SYSTEM DOWN AT 17.00, and press F6, the following command is generated and entered:

```
MSG USER1 SYSTEM DOWN AT 17.00
```

Note: The plus sign (+) leaves a blank after the text before concatenating it with the entered string.

A suffix function key acts like a prefix key, but adds its value to the end of the line of text where your cursor is positioned (that is, the command line or any other line in the OCS window display area).

Assign Your Own Values to Function Keys

You can assign your own values to function keys so that they invoke an NCL procedure or act as the Enter key. If a function key is being used as the Enter key, you can redefine the Enter key to perform an OCS function.

Specify Commands to Function Keys

To redefine function keys, use the PF command. You can specify the new function key value as *one* of the following:

Conversational

The value of the function key appears in the command line so that it can be edited before being issued.

Immediate

The function key performs an immediate function such as Enter.

Suffix

The function key value is placed at the end of an entry in the command line.

Prefix

The function key value is placed at the beginning of an entry in the command line.

Example: Assign a Conversational Function Key

To assign the SHOW NCL command as a conversational function key to the F4 key, enter the following command:

```
PF4 CONV,SHOW NCL=
```

When you press F4 from now on, SHOW NCL= is displayed at the command prompt so that you can add to it before executing it.

Example: Define Multiple Commands

You can use a semicolon as a command separator in the function key value to define multiple commands. When entering the PF command, specify two semicolons.

To define F20 to clear the screen and display users, enter the following:

```
PF20 CLEAR;;SHOW USERS
```

Set Function Keys as Enter Keys

The PF command can set a function key to act as the Enter key. The Enter key is defined as an immediate function key with no associated value.

Example: Define Enter Key

To define PF12 as the Enter key, enter the following:

```
PF12 IMM
```

PF12 acts as the Enter key because there is no entry after IMM. When you press F12, text is executed from the command line as if the Enter key is pressed.

Redefine the Enter Key

You can use the ENTER command to redefine the action of the Enter key.

Important! You must define at least one function key to act as Enter *before* you redefine the Enter key.

Example: Redefine Enter Key to Act as CLEAR Command

To redefine the value of the Enter key to act as the CLEAR command, type the following:

```
ENTER CLEAR
```

The Enter key no longer acts in its standard manner.

Example: Reset Enter Key

To reset the Enter key, type the following text and then press the function key that is defined as Enter:

```
ENTER IMM
```

Specify Function Keys Using NCL Procedures

You can also assign values to a function key from an NCL procedure. By setting the appropriate function keys in an NCL procedure and setting your initial command to execute the NCL procedure on entry to OCS, you can set the function keys for your OCS window.

Use Commands in OCS

OCS windows can be used to enter product commands and monitor the results. Commands are entered on the command line and take effect when you press the Enter key.

You can access a list of all product commands from online help.

To access the list of commands from any OCS window

1. Press F1 (Help).

The OCS Overview panel appears.

2. Enter **S** beside the List of Commands topic.

The commands are listed. You can select any of the commands displayed to get more information about its use and syntax.

Command Authority Levels

All commands are assigned an authority level within the range 0 to 255, zero being the lowest and 255 the highest authority level. The operands on some commands might require a higher authority than the base command itself.

You are allocated a command authority level in your user ID definition, corresponding to the scope of system control you require. Whenever you enter a command, your user ID authority level must be equal to or higher than the authority level of the command entered, otherwise the command is rejected.

This authority level checking also applies to commands executed from NCL processes invoked under your user ID.

Abbreviate Commands

All commands consist of a single command, which can be followed by one or more operands. Most commands can be abbreviated to the smallest number of characters consistent with their being distinguishable from any other product command. For example, the SHOW command can be abbreviated to SH.

Concatenate Commands

Several commands can be entered simultaneously by concatenating them into the same OCS command line and separating each command with a semicolon (;). The concatenated commands are processed from left to right in the order they are entered.

Example: Concatenate Commands

The command string `D LU10;D LU11` is treated by the system as two independent commands:

```
D LU10
D LU11
```

You can use the `CLEAR` command in this manner to clear the display area before the results of the next command are displayed. For example:

```
CLEAR;D BFRUSE
```

If you need to enter a semicolon as part of a command (that is, to use it as part of the command text), you must enter two semicolons instead of one.

To enter the command `a;b`, you must enter `a;;b`. The second semicolon is stripped from the text and the `a;b` string passed to the system as a single command. The remaining semicolon is not regarded as a command separator character.

Command separators are specified by using the `PROFILE CMDSEP` command.

Prevent Command Concatenation

You can prevent command concatenation by using the `CMDSEP` operand of the `PROFILE` command. When `CMDSEP` is set to `NO`, semicolons are not regarded as command separators and are always treated as part of the command string.

You can assign concatenated commands to function keys because the value of the `CMDSEP` operand is overridden by the value that the operand contained when the function key was defined.

Reuse Commands

If you enter a command regularly, you do not need to retype it every time you want to issue the command. There are facilities provided with OCS that let you reuse commands you have previously entered.

More information:

[Use the Command Stack](#) (see page 55)

[Retain Commands on the Command Line](#) (see page 55)

[Copy Display Lines into the Command Line](#) (see page 55)

Use the Command Stack

Each OCS window keeps a stack of the commands most recently entered from its command line. The stack does not include immediate function key entries. The number of entries kept in this stack can be changed by using the PROFILE CMDSTACK command.

You can use the command stack to retrieve previous commands entered and redisplay them on the command line so that they can be modified for re-entry.

Commands are retrieved from the stack using the CS+ or CS- commands. The default system function key series includes settings for the CS+ and CS- commands. These are F10 and F11 respectively. We recommend that you retain these.

Retain Commands on the Command Line

When you execute a command, the command can be retained on the command line so that you can execute it again, or edit the command before executing it again. This facility lets you increment and enter command sequences with minimal effort.

This feature can be turned on or off using the PROFILE CMDKEEP command. When turned off, the command line is cleared as soon as the Enter key is pressed and a command must be retrieved from the command stack if it is required again. When turned on, the command you enter is retained on the command line so that you can enter it again.

Copy Display Lines into the Command Line

To copy a command (or some other message) from an OCS window display area to the command line, put the cursor on the line you want to copy, and enter CS+ or CS- (or press F10 or F11).

The command or message appears in the command line.

Rename Commands

EQUATE commands can be included in initialization procedures to do the following:

- Override or rename standard commands
- Define a series of 1- to 8-character strings for use in place of lengthy command strings

Monitor and Control in OCS

OCS allows you to monitor and control your regions by receiving messages and allowing you to issue commands. Events from your network are sent to your OCS window. You can issue commands to take control of any problems that might occur.

As you receive messages and output from commands, you can control, reorder, or clear output on the screen so that it can be read more easily.

Control Message Presentation Speed

When the bottom line of the display area is filled, the system pauses before wrapping back to the top of the display area to write the next message.

Sometimes, a large number of messages might be sent to the screen within a very short period of time. This causes the display to roll messages faster than you can read them. There are two options you can use to temporarily suspend message delivery or change the way the messages display:

- The HOLD option
- The AUTOHOLD option

Stop Message Flow Manually

To stop the flow of output to the screen at any time, press the Enter key while nothing is in the command line. This freezes the display and no further messages appear until you enter data.

While the screen is frozen, the word HOLDING appears immediately above the command line.

Stop Message Flow Automatically

The default value for automatic hold supplied with your system automatically freezes an OCS window when a message fills the last line and there are messages queued to wrap back to the top of the screen. This is specified by the AUTOHOLD command.

When AUTOHOLD freezes your screen, the caption AUTOHOLD is displayed above the command line. No further messages appear until you input something.

The AUTOHOLD command option is part of your operator profile.

Note: If more unsolicited messages arrive while the screen is in HOLDING or AUTOHOLD mode, the caption above the command line changes to MSG QUED, and the terminal alarm sounds.

Message Queue Holding Limit

Your system queues a limited number of messages for an OCS window while in the HOLDING or AUTOHOLD mode. The queue limit default before any OCS window messages are discarded is 200 messages.

The HOLDING or AUTOHOLD caption above the command line changes to 75% LIMIT, HOLD LIMIT, and then MSGS LOST, as this limit is approached, reached, and then exceeded. Each caption change also sounds the terminal alarm. These conditions vary and update while you actively monitor and release system messages in the OCS window.

You can define the queue limit for each OCS user window by using the PROFILE command.

Contention Delay Interval

One of the characteristics of an OCS window is that your system can send messages to your window at the same time as you are entering a command. These messages are displayed differently depending on the type of terminal you are using:

A Non-SNA Terminal

Any data you have just entered is immediately frozen and any new data entered is ignored while the message writes to the screen. You can then continue to type in your command text when message delivery has finished.

An SNA 3270 Terminal

A contention condition arises. The terminal is seen as being in a send state (because you have started typing on the keyboard), and refuses to accept any output from your system until your input has been sent. However, rather than defer your system, the system interrupts you after a set period and forces the output of a message.

The default contention delay interval is 15 seconds. This is usually long enough to let you complete a standard command input operation.

Unwrap Messages

To resequence or unwrap messages displayed in your OCS window, enter the **ORDER** command.

The OCS messages are redisplayed in the window in chronological order, with the oldest messages at the top of the window.

The ORDER command is assigned to F12 by default.

Note: This command does not affect the HOLDING or AUTOHOLD condition.

Clear the OCS Window

After many messages have appeared in your OCS window, you may want to clear the window before any new messages arrive.

To clear your OCS window, enter the **CLEAR** or **K** command.

Receive Non-roll Delete Messages

Most messages displayed on an OCS window are classified as roll delete messages. This means they are displayed once and eventually roll off the top of the screen as subsequent messages arrive and overwrite them.

When a non-roll delete (NRD) message is delivered to an OCS window, it remains in your OCS window until deleted. The NRD messages are in two categories:

- Those that are remembered by the system and are retained until explicitly deleted by the issuing process
- Those that are only displayed at individual OCS windows until deleted and are not remembered by the system

The non-roll delete area is separated from the roll delete area by a delimiter line. This line is usually a series of dash (-) characters. To change the character, use the PROFILE NRDELCH command.

NRD messages are managed centrally and held in a queue. Your system administrator determines the size of this queue. If there are more NRD messages than this limit, the oldest outstanding NRD message is deleted to remove copies of the message from all affected OCS windows.

However, NRD messages from &WRITE NCL statements are never deleted automatically. It is therefore only possible to exceed the NRDLIM queue depth if large numbers of &WRITE-generated NRD messages exist at the same time.

A warning message is sent to all OCS users with monitor status to notify them when the NRD message queue reaches 75 percent full.

You can hide these messages to allow other message flows to continue in your OCS window, and then reveal them again when you are able to deal with them.

Hide NRD Messages

To remove an NRD message from the OCS display, move your cursor to the line on the screen with the NRD message you want to remove and then press Enter.

The NRD message disappears and the screen is reformatted. Removing NRD messages in this way provides more room for pending NRD messages or a larger roll delete area.

Note: System NRD messages are not deleted from the NRD message queue, only from your OCS window. NRD messages specific to your OCS window are deleted, and cannot be recalled.

Restore Hidden NRD Messages

Hidden NRD messages can be restored by entering the NRDRET command. The oldest hidden NRD messages are returned to the non-roll delete area first, until the area has expanded to its maximum size.

The NRDRET command displays all hidden NRD messages that you are entitled to view, including those that occurred before you entered OCS and any that are still outstanding.

NRDRET can be issued from any environment capable of receiving NRD messages, including NCL &INTCMD environments.

Delete NRD Messages

An NRD message is automatically deleted when *one* of the following conditions is satisfied:

- The condition to which an NRD message refers is satisfied
- An NCL process issues an &NRDDEL NCL statement to delete a specific NRD message
- The NCL process that generated the NRD is terminated

You can only delete NRD messages that are remembered by the system by using the PURGE command.

Use NRD Messages with ROF Sessions

Messages that originate from a remote system carry the NRD message attribute and appear as NRD messages, in the same way as locally-produced messages.

When an INMC link fails and breaks any ROF sessions traveling across it, all NRD messages from that remote system are automatically deleted.

When you close a ROF session to a particular remote system by using the SIGNOFF command, any NRD messages you have received across the ROF session are deleted from your window. Other users displaying the same NRD messages are not affected.

Use the Activity Log to Help Monitor Your Regions

The activity log records all commands, responses to commands, and messages that occur in your regions. By accessing the activity log when you are in OCS you can browse through recent activity on the system to assist you in locating information and analyzing problems.

To access the activity log browse function from OCS, enter **/LOG** at the prompt.

On initial entry to the activity log, you are positioned at the end of the log for the current day. You can use the F8 (Forward) and F7 (Backward) function keys to scroll through the log for the current day as well as for previous days.

Note: For more information about locating information in the activity log, press F1 (Help) from the activity log panel.

Record Additional Information in the Activity Log

You can use the LOG command to record additional information in the log that is outside the scope of that recorded by default. For example, enter the following at the prompt:

```
LOG PRODUCTION LINK TO LOS ANGELES DOWN FOR MAINTENANCE
```

This produces the following entry in the log for the time the command was issued:

```
LOG ENTRY : PRODUCTION LINK TO LOS ANGELES DOWN FOR MAINTENANCE
```

You can record a message up to 256 characters long.

Interpret Messages and Codes to Help Monitor Your Region

The information database provides categories of information about commonly used codes and errors. By accessing the information database from OCS, you can get information about error messages that appear in your OCS window.

To access the information database from OCS, enter **/CODES** at the prompt.

Issue Commands

Being able to issue commands from OCS is an important part of controlling your regions. From OCS you can issue commands to the background processes of your product, and you can set commands to issue automatically, based on a specified time.

Issue Commands in Background Environments

Background environments are internal to your system and services. They process commands submitted to them by users and support system level procedures such as LOGPROC. Each background process has a user ID, but is not associated with any physical terminal.

The following background environments are available:

BSYS

Background system environment

BMON

Background monitor environment

BSVR

Background server environment

BLOG

Background logger environment

You can send commands to these environments for them to execute, as if they were real OCS users by using the SUBMIT command. You can submit commands or NCL procedures. For example, if you want the background system environment to start the procedure MONPROC, enter the following command:

```
SUBMIT BSYS START MONPROC
```

After a command is submitted, its processing is managed by that environment. It is not affected if you log off or leave OCS, and its command authority remains the same as the user ID of the submitter.

Background environment processing is ideal for monitoring an NCL procedure that regularly checks the status of network components. Commands directed to the Background Monitor route the command and its results to all monitor status terminals logged on to the system, and to the activity log. Commands directed to the Background Logger for execution log the command and its results only.

[Timer commands](#) (see page 62) can also be routed to background environments by the SUBMIT command or by the ROUTE operand for the timer command being issued.

Issue Commands at Specified Times

You can issue commands at specified times and at specified intervals. These commands are known as timer-initiated commands. The following timer-initiated commands are available:

AT

Executes commands at a specified time of day. Timer commands use a 24-hour clock with the format *hh.mm.ss*.

Limits: 24.00.00 (midnight)

EVERY

Repeats commands at a given time frequency.

Default: 10 seconds

Timer commands can be entered in OCS, or included in NCL procedures.

A maximum of 9999 concurrent timer commands is supported, and this maximum is the default.

If you log off after issuing a timer command, that command is not executed. However, you can use the ROUTE or KEEP operand when you enter an AT command to specify another user to issue the command in your place. This feature allows you to sign off and have the results of the command returned to you when you sign on again.

The ROUTE and KEEP options are ideal if you are including timer commands for specific operators in the system initialization procedures that are executed automatically during startup.

Timer commands can also be specified with a limit to the number of times they can execute before being automatically purged.

Example: Monitor Users at a Specified Interval

To monitor the users that are logged on to the system every half hour, enter the following command:

```
EVERY .30 CMD=SHOW USERS
```

Also, if you want to remind users of a three o'clock meeting one hour before it starts, enter the following command:

```
AT 14.00 MSG ALL DON'T FORGET MEETING AT 15.00
```

When a timer command executes, the command text is echoed on all applicable terminals as if the command had been entered from those terminals. A unique timer ID prefixes the command text echo and has the following format:

```
#nnnn command_text
```

Display Active Timer Commands

You can display pending timer commands by using the SHOW TIMER command. By default, this command lets you display any timer commands initiated by your user ID. However, by specifying the ALL operand you can display all outstanding timer commands on your system.

Example: Display Active Timer Commands

To find out what timer commands you have initiated, enter the following command:

```
SHOW TIMER
```

Using the example given above, the following is displayed:

```
ID BY INTERVAL -USERID-R LIM CNT K/P ENV P/M TID  NEXT
4 EV 00:30:00 USER01      0  0 NO PRI YES -   12:29:48
  CMD=SHOW USERS
5 AT 14:00:00 USER01      0  0 NO PRI YES -   14:00:00
  CMD=MSG ALL DON'T FORGET MEETING AT 15.00
NUMBER OF TIMER COMMANDS DISPLAYED WAS 2.
```

Delete Timer Commands Manually

When you initiate a timer command, the system allocates a unique four-digit number known as the timer ID, or purge ID. This number prefixes all displays resulting from that command, and must be used when manually deleting a timer command.

To delete a timer command manually, use the **PURGE** command.

To delete a timer command created by another user, you require a command authority level of 2 or higher.

Example: Delete Command

To delete an AT timer command, enter the following command:

```
PURGE TIMER=5
```

The value 5 is the purge ID assigned to the AT command.

Delete Timer Commands Automatically

By default, your timer commands remain active only while you are logged onto the system. Before each attempt to execute the command, the system checks that you are still logged on.

If you are no longer logged on to your system, the timer command is automatically deleted, without further execution.

Redirect Timer Commands

If you want your timer-initiated commands to continue to execute after you log off, you can redirect the command results to the background logger, background monitor, or the system background environment.

To redirect the timer command, specify an AT or EVERY command with the KEEP operand.

By default, the KEEP operand requires a command authority level of 2 or higher.

Example: Redirect Timer Commands

To redirect the SHOW USERS command for execution by the background system environment, enter the following command:

```
EVERY .30 KEEP=SYS CMD=SHOW USERS
```

If the KEEP operand is in use, the execution of timer commands continues irrespective of whether you are logged on to the region.

Limit Timer Command Executions

When defining a timer command, you can use the LIMIT operand to specify a limit on the number of times the command is executed. When this limit is reached, the command is automatically purged.

The limit you assign and the number of times a command has already executed are displayed by the SHOW TIMER command.

Example: Limit Timer Command Execution

To limit the number of times the SHOW USERS command is executed to 5, enter the following command:

```
EVERY .30 LIMIT=5 CMD=SHOW USERS
```

When the SHOW USERS command has been executed five times, the timer command is deleted.

Execute a Timer Command Under Another User ID

The ROUTE operand lets you direct a command for execution under another user ID—the target user ID. The operand requires a command authority level of at least 2.

With this option, the timer command is retained even if the target user ID is not logged on. Command execution is bypassed and the time interval reset. The command is attempted again only after the time interval has again elapsed.

Example: Execute Timer Command Under Another User ID

If you want USER02 to execute the SHOW USERS command, enter the following command:

```
EVERY .30 ROUTE=USER02 CMD=SHOW USERS
```

Specify Concatenated Commands in Timer Commands

Concatenated commands can be specified in the command text for a timer command. Separate each command in the concatenation with a colon (:). These are internally translated into normal concatenation characters, that is, semicolons (;), before execution.

Execute or Start NCL Processes from OCS

There is an NCL processing environment for each window of your terminal that allows commands and NCL processes to execute on behalf of that window.

When you use an EXEC or START command to invoke an NCL process, the NCL process executes in the NCL processing environment for the OCS window.

Note: If you enter the EXEC or START command incorrectly, the system attempts to execute the command as if it were an NCL process.

Any NCL process can have a dependent processing environment that lets it issue commands or execute other NCL processes independently using the &INTCMD statement. NCL procedures can also use ROF sessions to collect information from other systems.

Execute NCL Processes Serially

An OCS window can execute a serial stream of NCL processes so that they are invoked one after the other. Serial execution is suitable for processes with a short duration.

To execute NCL processes serially, use the **EXEC** command.

Processes invoked by the EXEC command can issue the &PAUSE statement to wait for further input from the OCS window. The GO, END, FLUSH, and INTQ commands, together with the process's unique identifier, let you communicate with the process.

Example: Execute Processes in Sequence

To execute PROC1 and PROC2 in sequence, enter the following commands:

```
EXEC PROC1  
EXEC PROC2
```

Your OCS window places the two processes in an EXEC queue, which are executed on a first-come, first-served basis. Process PROC1 is scheduled for immediate execution and process PROC2 is queued to execute after PROC1 ends.

Execute NCL Processes Concurrently

An OCS window can execute NCL processes in parallel at the same time.

To execute NCL processes concurrently, use the **START** command. If you enter the name of an NCL procedure by itself, the START command is implied.

Any started procedure can issue an &PAUSE statement to wait for further input from GO, END, and FLUSH commands from the OCS window. These commands, together with the unique identifier for the process, let you communicate with the process explicitly.

Example: Execute NCL Processes Concurrently

To execute PROC1 and PROC2 at the same time, enter the following commands:

```
START PROC1  
START PROC2
```

NCL Identifiers

Each NCL process is allocated a unique identifier that links it to the issuing OCS window. This ensures any &WRITE or &PANEL statements issued by the NCL process (or any other processes it starts or executes), are returned to that window only. If the window is terminated, any queued process is deleted.

Execute an NCL Process from a Serial or Concurrent Process

An NCL process executed from an OCS window (or any process it invokes) can itself issue EXEC or START commands.

If an EXEC command is used to execute an NCL process, the process issuing the command is suspended when the new process starts executing. Only when the new process ends does the issuing process resume processing.

Invoking a process from another process in this way is called nesting. Nesting is an easy way to structure a series of processes.

Note: The &CALL PROC NCL statement is the recommended method for nesting procedure calls.

If a START command is used to execute an NCL process, the new process starts executing immediately. The new process runs concurrently with the invoking process and independently of it. Each process is unaffected by the termination of the other process.

Advantages of Started Procedures

Using the START command to invoke NCL processes has the following advantages:

- You can perform relatively complex, long-term tasks from your OCS window. This does not prevent other operations from performing concurrently.
- You can perform periodic checking of the network status without operator involvement.
- You can operate a large number of independent, slave procedures on behalf of one OCS window. This lets you monitor many different aspects of the same operation, and various procedures need only communicate with you if errors are detected.

Chapter 7: Using Commands to Control Transmissions

This section contains the following topics:

[Use of Commands](#) (see page 69)

[PROFILE Command](#) (see page 69)

[TRANSMIT Command](#) (see page 69)

[FTSMOD Command](#) (see page 70)

[FTSINIT Command](#) (see page 70)

[SHOW FTS Command](#) (see page 70)

[Timer Commands](#) (see page 71)

[Batch Jobs](#) (see page 71)

Use of Commands

For ease of use, CA SOLVE:FTS functions are provided through menus and panels. However, under some circumstances, it is advantageous to request those functions using commands. You can issue commands from an OCS window, the system console, within NCL procedures, or batch jobs.

PROFILE Command

The PROFILE command supports the FTSMMSG operand. Authorized users receive messages generated by CA SOLVE:FTS when they are profiled with FTSMMSG=YES and UNSOL=YES. The FTSMMSG operand can be used to suppress receipt of messages if they are not required. As with other PROFILE operands, the FTSMMSG operand applies only to the OCS window from which the command is issued.

Note: For information about the syntax of this command, see the online help.

TRANSMIT Command

The TRANSMIT command (or its pseudonym XMIT) can issue one or more transmission requests. It can be issued by any user with request privilege. Normal private, system, and access mask checking applies. The TRANSMIT command can be used from an NCL procedure that is being executed by an authorized user.

Note: The Background Monitor and Logger regions cannot execute the TRANSMIT command. For information about the syntax of this command, see the online help.

FTSMOD Command

The FTSMOD command enables the OCS user to modify the status of a transmission request. This includes:

- Holding or releasing the request
- Acknowledging the request
- Restarting it after an error

FTSMOD can be executed from within an NCL procedure to automate the supervision of CA SOLVE:FTS activity.

Note: A user can only modify those transmission requests for which they have control privilege. For information about the syntax of this command, see the online help.

FTSINIT Command

The FTSINIT command modifies the status of CA SOLVE:FTS initiators. You can use it with the SHOW FTS INIT command to control initiator settings dynamically from an OCS window or from an NCL procedure.

Note: For information about the syntax of this command, see the online help.

SHOW FTS Command

The SHOW FTS command displays the status of initiators and transmissions.

To display the status of the initiators for a destination, enter the following command:

```
SHOW FTS INIT=destname
```

You can execute the SHOW FTS command internally by using &INTCMD from an NCL procedure and use the results to produce a tailored full-screen panel.

Note: For information about the syntax of the SHOW FTS command, see the online help. For information about &INTCMD, see the *Network Control Language Reference Guide*.

Timer Commands

The TRANSMIT command can be executed as a timer-based command, so that a user can request transmission of a data set at a particular time of day or at a specified frequency. A timer-based FTSINIT command can be used to enable or disable transmission to a destination, or start or stop initiators at a given time of day.

More information:

[Issue Commands at Specified Times](#) (see page 62)

Batch Jobs

The interaction between CA SOLVE:FTS and batch jobs to create data sets for transmission or process-transmitted data sets is best accomplished by using the Batch Command Interface (BCI).

Note: For more information, see the *Network Control Language Programmer Guide*.

Chapter 8: Using the TSO Interface

This section contains the following topics:

[TSO Interface](#) (see page 73)
[Subcommand Environment](#) (see page 74)
[Get Help in the Subcommand Environment](#) (see page 75)
[NMCMD FTSDEF Subcommand](#) (see page 76)
[NMCMD FTSDEL Subcommand](#) (see page 82)
[NMCMD FTSEND Subcommand](#) (see page 83)
[NMCMD FTSXMIT Subcommand](#) (see page 84)
[TSO Interface NCL Procedures](#) (see page 84)
[Messages Returned to TSO](#) (see page 86)
[Return Codes from \\$USTSFTS](#) (see page 89)
[Examples of Requests Using NMCMD](#) (see page 89)

TSO Interface

TSO users can use the supplied [NMCMD command processor](#) (see page 74) to enter requests directly from TSO. The NMCMD command processor provides a general interface between TSO and the NCL procedures.

After invoking the NMCMD processor, various subcommands are available. All subcommand functions are initially edited by TSO subcommand processors before being forwarded to where the standard CA SOLVE:FTS NCL Interface procedures are called to process the requests. This lets TSO users request file transmission activity without connecting to the CA SOLVE:FTS region directly.

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

Subcommand Environment

The NMCMD command processor can be invoked during your TSO session to perform online requests, or from a batch job by executing TSO in the background. The primary function of the NMCMD processor is to contact the specified region 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 the region. When this session is established, any of the NMCMD subcommands can be entered for processing.

After entering a subcommand, the request is forwarded to the NCL procedure \$USTSFTS running under User Services. This procedure creates the &FTS variables required for each request and calls the appropriate NCL procedure to perform the function.

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

Note: Functions invoked by NMCMD subcommands execute the CA SOLVE:FTS NCL Interface procedures. Hence any restrictions imposed by these procedures (other than those imposed by CA SOLVE:FTS itself) also apply to these TSO subcommands.

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

Get Help in the Subcommand Environment

The subcommand environment in TSO provides online help on NMCMD and its subcommands.

To get help about NMCMD, enter the following command from TSO:

```
H NMCMD
```

To get help about an NMCMD subcommand, enter the following command from TSO:

```
NMCMD region_applid [USER(user_id/password)] CMD('HELP subcommand')
```

region_applid

Specifies the application ID of the region to which you want to issue a subcommand.

user_id/password

(Optional) Specifies the credentials you want to use to log on to *region_applid*.

Default: Your TSO user ID and password

subcommand

Specifies the subcommand for which you require help.

Important! To log on to a region successfully from TSO, the logon user ID must be defined with TSO Autologon = Y.

NMCMD FTSDEF Subcommand

The FTSDEF subcommand lets TSO users create and store CA SOLVE:FTS definitions in the VFS database. No file transmission activity results from the definition, but a subsequent transmit request can be used to request the transmission.

```
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( torecfm ) ]
            TOBLK( toblksize ) ]
            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 ]
        [ PRIVATE | SYSTEM ]
        [ FRCMD( 'from command' ) ]
        [ TOCMD( 'to command' ) ]
        [ GROUP( groupid ) ]
        [ UDATA( 'user data' ) ]
        [ FRXMIT( ftsdef ) ]
```

NAME(*ftsname*)

(Mandatory) Provides the name for 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 in the region to which you are connected, otherwise, the request will fail. As distributed, the CA SOLVE:FTS procedures restrict all such names to only those beginning with the user's user ID, regardless of the user's privileges. Check with your installation representative or system administrator for any name restriction changes.

COPY(*ftsdef*)

(Optional) Provides the name of an existing transmission definition to copy. If used, the only other operands that 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' data set*, and *'From' data set* respectively because those operands are mandatory.

Note: The COPY operand should be preceded only by the NAME operand because any intervening operand values are lost.

DEST(*ftsdest*)

(Mandatory) Provides the name of the destination where this transmission is to be sent.

Limits: 1 to 12 characters

CLASS(*ftsclass*)

(Optional) Provides the 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*)

Specifies the name of the 'From' or source data set to be transmitted to the target region. This can be any valid data set 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 data set prefixing, the first level qualifier can be omitted because it is automatically inserted by the command processor, *unless* the data set name is enclosed in quotation marks. If the data set is not cataloged, FRVOL and FRUNT operands must be present.

FRDSP(OLD | SHR)

Specifies the disposition of the *fromdsn* data set.

FRVOL(*fromvol*)

Specifies the volume serial information of the *fromdsn* data set if it is not cataloged. If this operand is specified, the FRUNT operand must also be present.

FRUNT(*fromunit*)

Specifies the unit information (for example, TAPE, SYSDA, and so on) of the *fromdsn* data set if it is not cataloged. 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 data sets. This operand is mutually exclusive with the specification of a member in the *fromdsn* data set name. Up to 99 members may be present in the list. Any members can be renamed in the target data set 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 is transmitted in a separate transmission. The *ftsname* provided by the NAME operand should not exceed 10 characters if the MEMBERS operand is used because each actual request after the first one generates a name consisting of the *ftsname* suffixed by two digits (02-99).

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

(Mandatory) Receives the transmitted file. If the name specified is 'SYSOUT', the *fromdsn* data set is transmitted directly to the JES spool. In this case, no other operands that begin with TO should be present, but those beginning with SO may provide any required SYSOUT options. If a *todsn* data set name is supplied, it must be compatible with the *fromdsn* according to the usual requirements.

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

Provides the disposition of the 'To' or target data set. Unless NEW or REP is specified, then the *todsn* entered must be a cataloged data set at the destination. Otherwise, the TOVOL, TOUNT, TOSPC, TOLRL, TOBLK, and TORFM operands must provide the new data set information. The REP option is treated as NEW except that the data set is *deleted* and *replaced* if it exists already.

TOVOL(*tovol*)

Provides the volume serial information of the *todsn* when TODSP is NEW or REP. The new data set is created and cataloged on this volume. If this operand is omitted for a new data set, then the TOUNT operand must be specified and the destination will create the data set on any suitably authorized volume for the *tounit*.

TOUNT(*tounit*)

Provides the unit information of the *todsn* when TODSP is NEW or REP. If it is not supplied for a new data set, 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 data set 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 that 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.

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.

TOBLK(*toblksize*)

Provides the block size as a 1 to 5 digit number for new *todsn* data sets 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.

TOLRL(*tolrecl*)

Provides the logical record length as a 1 to 5 digit number for new *todsn* data sets 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.

TIMCO(Y | N)

Indicates whether an image-copy transmission is requested. If Y is specified, the DCB attributes of the *todsn* are 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*, *toblksize* and *torecfm*) in which case it is set to N.

SOCLS(*class*)

(Optional) Provides the SYSOUT class if TODSN(SYSOUT) is specified, and defaults to A.

SOCOP(*copies*)

(Optional) Provides 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 data set is to be placed in HOLD when TODSN(SYSOUT) is specified.

SODST(*dest*)

(Optional) Provides 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*)

(Optional) Provides the 1 to 4 character SYSOUT forms name if TODSN(SYSOUT) is specified.

SOFCB(*fc*b)

(Optional) Provides the 1 to 4 character SYSOUT FCB name if TODSN(SYSOUT) is specified.

SOUCS(*ucs*)

(Optional) Provides the 1 to 4 character SYSOUT UCS name if TODSN(SYSOUT) is specified.

SOPGM(*program*)

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

MSG | NOMSG

Indicates whether returned messages are written to the TSO terminal using PUTLINE macros. Error messages about session I/O cannot be suppressed by this operand.

PRIVATE | SYSTEM

Determines whether this transmission request is a PRIVATE or SYSTEM request.

Default: PRIVATE

Note: The user's 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 are allowed (in quotation marks if there are embedded blanks). Any product 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 are allowed (in quotation marks if there are embedded blanks). Any product command valid for entry from this source can be entered, including the EXEC and XMIT commands. Normal command authority checking applies.

GROUP(*groupid*)

Provides the 1- to 8-character user group identifier for this request. This information is propagated with the file transmission request for the benefit of the data set access authorization 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 (in quotation marks if embedded blanks are present) of user data that is associated with the request. This information is propagated with the file transmission request to the data set access authorization 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 definition that is to be chained for transmission from the current one after its successful completion. Normal privilege checking applies.

NMCMD FTSDel Subcommand

The FTSDel subcommand lets TSO users delete existing CA SOLVE:FTS definitions from the VFS database.

```
FTSDel  NAME( ftsname )  
          [          NOCHAIN | CHAIN ]  
          [          MSG | NOMSG ]
```

NAME(*ftsname*) (Mandatory)

Provides the name of the definition that is to be deleted.

Note: The CA SOLVE:FTS procedures only allow deletion of definition names that begin with the requesting user's user ID, regardless of their privileges.

NOCHAIN | CHAIN

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

MSG | NOMSG

Indicates whether returned messages are written to the TSO terminal using PUTLINE macros. Error messages about session I/O cannot be suppressed by this operand.

NMCMD FTSEND Subcommand

The FTSEND subcommand provides TSO users with an on demand 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( tount ) ]
            [ TOSPC( units pri sec dir ) ]
            [ TORFM( torecfm ) ]
            TOBLK( toblksize ) ]
            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 subcommand services on demand file transmission requests by performing the following in sequence:

FTSDEF

Creates a file transmission definition (or chained series of definitions) in the VSF data set in the region with which the NMCMD TSO program is communicating.

FTSXMIT

Requests transmission of the previously defined request (or request chain).

These functions are performed by invoking the relevant NCL procedures; therefore, any restrictions imposed by these procedures also apply to requests originating from TSO.

The operands for the FTSEND subcommand are identical to those for the [FTSDEF subcommand](#) (see page 76).

NMCMD FTSXMIT Subcommand

The FTSXMIT subcommand lets TSO users request that an existing CA SOLVE:FTS definition be transmitted. The definition can be added through the FTSDEF subcommand or through the normal CA SOLVE:FTS options.

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

NAME(*ftsname*) (Mandatory)

Provides the name of the definition to transmit.

MSG | NOMSG

Indicates whether returned messages are written to the TSO terminal using PUTLINE macros. Error messages about session I/O cannot be suppressed by this operand.

TSO Interface NCL Procedures

\$USERSER

Specifies the standard User Services procedure distributed with the region. It detects a logon from the External Interface through NMCMD and passes control to \$USERTSO.

\$USERTSO

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

\$USTSFTS

Extracts the data received from the TSO command processors and then calls the appropriate CA SOLVE:FTS NCL Interface procedure to perform the actual function requested. Called from \$USERTSO when a request is received for each of the subcommands of NMCMD. By default, procedure \$USTSFTS specifies conditional definition deletion for all CA SOLVE:FTS definitions added. Processing depends on the actual request, as follows:

- **FTSDEF**

Primes all &FTS variables with the values defined to the subcommand 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 is 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 &FTSFRCMD, &FTSTOCMD, and &FTSFCXMT are placed only 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 indicates 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 indicates the status of the request.

Messages Returned to TSO

The distributed procedures can return the following messages to the TSO environment following the execution of each subcommand. If the subcommand used specified the NOMSG option, then these messages are suppressed and not written to the terminal.

Note: Some messages are CA NetMaster messages that are received by the procedure following the execution of an internal command. Others are generated by the CA SOLVE:FTS procedures.

PFTS01

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

This message indicates a severe file error and provides the access as READ or UPDATE, the file key 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 indicating that the received member list specified more than 99 members. Only the first 99 members are processed, the rest are ignored.

PFTS03

NO MEMBERS FOUND WHEN MEMBER LIST SPECIFIED

This message indicates a request to process a member list was received but the member list was empty. The procedure terminates with a return code of 24.

PFTS04

TRANSMISSION REQUEST *name* IN ERROR

This general message indicates an error and is 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 a return code of 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 a 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 a return code of 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 that 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 that follows the successful addition of a single CA SOLVE: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 exists. The request is rejected and the procedure terminates with a return code of 16.

PFTS12

DEFINITION *name* DELETED

This informational message indicates the successful deletion of the named definition.

PFTS13

DEFINITION *name* NOT DELETED, RECORD NOT FOUND

A request to delete the definition named failed because 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 provided definition name 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 is 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 increased by 500 (producing a number in the range 500 to 599), which becomes the actual return code of NMCMD on termination.

The following return codes are produced by \$USTSFTS. All nonzero return codes are accompanied by one or more [messages](#) (see page 86).

0

Request was processed successfully.

8

Attempt to supply a FRCMD or TOCMD (command for execution in the From or To region after a transmission completes successfully) for a private definition. The definition is rejected because this facility is restricted to system definitions.

12

An attempt to delete a definition that 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 because a definition of that name already exists.

20

An FTSDEF or FTSEND 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 Requests Using NMCMD

The following examples show how to use the NMCMD TSO command processor and the subcommands when requesting CA SOLVE:FTS services. Enter the commands at the TSO READY prompt or from an ISPF panel allowing entry of TSO commands. Responses returned to the terminal are shown in bold, and prompts are in italics.

Example 1

The following command connects to the region, NMTEST, requests an on-demand file transmission to region NMPROD, then disconnects and returns to TSO. Assuming the command is entered in ISPF, the termination displays the normal *** :

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

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

Note: You need not specify the input data set prefix if profiled from TSO to use a data set prefix. Only the required operands are shown in this example, all others assume the default values.

Example 2

In this example, NMCMD is invoked from TSO and two commands are 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 data set. The second command copies an existing definition but directs the output to the data set specified:

```
READY
      NMCMD NMNY

NMREADY
      FTSEND  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
      FTSEND  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 executes as a batch job and several commands input to the region 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 to input a list of commands to NMCMD in the foreground by using the READ subcommand of NMCMD.

Assume the following NMCMD subcommands are 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 can then be invoked in the following manner:

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

These commands can be used from a TSO CLIST and because all of the subcommands contain the NOMSG option, no output to the terminal results. The procedure can check the return code from NMCMD before sending a message to the invoking user.

Chapter 9: Using the ISPF Dialog Interface

This section contains the following topics:

[ISPF Dialog Interface](#) (see page 93)

[ISPF Dialog](#) (see page 94)

ISPF Dialog Interface

The ISPF Dialog Interface integrates into the existing ISPF environment and provides a means of effecting file transmissions.

Although the dialog contains some additional functions, it does not implement all CA SOLVE:FTS options and is not intended as a replacement for the standard facility.

Any CA SOLVE:FTS region in the network that is accessible from the TSO host can perform file transmissions. The choice depends on the location of the source data set. If the source data set resides on the TSO host system, then the transfer is from the *local* host and is affected by the region running on the same CPU. If the source data set is on another host system, then it is considered a *remote* host and the transfer is affected by the region running on that remote CPU.

Note: Certain functions are restricted to local host transfers and to specific data set types.

More information:

[Direct File Transmissions](#) (see page 96)

[Staged File Transmissions](#) (see page 96)

ISPF Dialog

Your systems programmer may have configured ISPF to provide the CA SOLVE:FTS ISPF Dialog Interface as a menu option.

On entry, a FROM panel presents details of the type of transmission request, the data set the user wants to transmit, and the system on which it resides. Valid options are:

C

Performs a *direct* copy from the source data set to the target data set. The source data set may be a sequential file or a PDS member. If the source system is the local system, the source data set may specify a PDS and a selection list is provided if no member name is supplied. If the source system is a remote system and no member name is supplied, the source data set is assumed to be a sequential file.

M

Performs a *direct* copy of multiple members of a PDS. The source data set 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 after TO details have been supplied.

S

Performs a *staged* copy from a local PDS or VSAM data set only. For a PDS data set 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 recognized as the local system, then the data set is checked to discover its organization (DSORG) and allocation details. If a new target data set is requested, the information is used to prime the panel prompting for the new data set allocation details.

Following the entry of *from* details, a TO panel requests the information about the target data set.

If option M is selected, the source and target data sets must be PDSs and no member names can be supplied because they are entered on a subsequent panel.

If option C is selected and the *from* system is the local system, the target data set is edited as follows:

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

If option C is selected and the *from* system is a remote system, the target data set is edited as follows:

- If a member name is specified on the FROM panel, the data set is assumed to be a PDS, otherwise it is assumed to be a sequential file. If a member is specified on the target data set it too is assumed to be a PDS.
- When no member is supplied in the target data set name but one was specified in the source data set name, if the area of the panel specifying an ISPF LIBRARY is used to describe the data set, 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 data set; otherwise, the data set is assumed to be a sequential file.

The target data set 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 appears.

For a staged copy of a VSAM data set, a panel prompts for information on whether VSAM record replacement is allowed, and for a limiting range of keys that should be copied (applicable to KSDS data sets only). If the target VSAM data set is specified as NEW, a panel appears requesting confirmation of the transmission before a temporary export is performed, the data set 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 data set that is a PDS on the local system, a member selection list is provided if no member name is specified on the FROM panel. Users can scroll through the selection list until all selections are made. Members can 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 can only be selected once. Any subsequent selections of the same member are ignored. Pressing the E key terminates the selection processing and cancels the request if no selections were made. To cancel the operation after having made one or more selections, use the CAN or CANCEL primary line command.

After all the information required to define the transmission is obtained, a panel appears 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 request to the appropriate system.

Any error messages occurring during the dialog, or any informational messages returned by the region following the successful scheduling of the request, are then displayed. Finally, control is returned to the FROM panel where you can repeat the process or press the E key to exit.

Direct File Transmissions

Data sets with formats that CA SOLVE:FTS can read can be used for *direct* file transfer. This means that the source data set 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 data set in the target system.

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

Staged File Transmissions

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

This technique allows data set types not usually supported for direct file transmission to be copied from one system to another. The dialog application provides such support for transmitting PDS load library members, entire PDS data sets, and VSAM data sets.

Staged file transfers are restricted to data sets from the local system and to existing target data sets only.

Edit Restrictions

When requesting a direct file transmission from a remote system, the dialog application cannot check all details about the source data set, or to even check whether the data set exists. In addition, it may make certain assumptions regarding the nature of the data set format, depending upon the request and context, and so on.

Chapter 10: Using Print Services

This section contains the following topics:

[Print Services Manager](#) (see page 97)

[Access PSM](#) (see page 97)

[List Entries in the Print Queue](#) (see page 98)

[Confirm Printing](#) (see page 99)

Print Services Manager

Print Services Manager (PSM) lets you control the physical printing of the reports your organization generates on JES or network printers. Output can be viewed online before or after printing and can be redirected to another destination.

PSM provides the following facilities:

Print Spooling

Writes output to a print spool providing more control over output. This facility lets you redirect output to another printer if one is not available.

Centralized Printer Definition Facilities

Supports VTAM (LU1) and JES (SYSOUT) devices and lets you assign printer aliases. This facility also allows the output destination to be a printer exit.

Print Request Control

Lets you hold, release, browse and delete print requests, redirect print requests to another printer, change priorities and numbers of copies, and display the status of requests.

Notes:

- For information about defining and maintaining printers, see the *Administration Guide*.
- References to JES also apply to VOS3's JES3 and JES4 subsystems.

Access PSM

To access PSM

1. Enter **/PSM** at the prompt.

The PSM : Primary Menu appears.

List Entries in the Print Queue

You can list all of the entries that are queued to print, and on which printer they are to print.

To display the entries in the print queue

1. Enter **Q** at the prompt on the PSM : Primary Menu.

The PSM : Output Queue appears.

Note: You can limit the display to the print queue for a specific printer by specifying a printer in the Printer field on the PSM : Primary Menu before entering the Q option.

Display the Output of a Print Request

You can preview the output to see exactly how the print request looks when printed.

Note: Only data lines, not heading lines, are displayed.

To browse the output of a print request, enter **B**, **/**, or **S** next to the required print request on the Output Queue panel.

The details appear.

Example: Browse Output

```

PROD----- PSM : Browse Output -----REQ# 0265
Command ==>                               Scroll ==> PAGE

S A B U Data
  --+---10--+---20--+---30--+---40--+---50--+---60--+---70
N N =====
1 N  COMMAND ENTRY CAPTURE PRINT
1 N =====
2 N  USERID : USER01             NAME : USER NUMBER 1      LO
2 N  DATE   : MON 26-APR-2010
2 N  TIME   : 11.16.39
2 N =====
2 N  pr
1 N  N10601 USERID: USER01 TERMINAL-ID: TERM02
1 N  N10602 NCL PROCEDURE LIBRARY ID: COMMANDS
1 N  N13450 PANEL SERVICES PATH NAME: PANELS
1 N  N10603 AUTHORITY LEVEL IS 82
1 N  N13451 NO EDS PROFILES ACTIVE IN ENVIRONMENT.
1 N  N13433 USER SERVICES PROCEDURE: $USERSER
1 N  N10624 NO NPF COMMAND RESTRICTIONS.
1 N  N10627 PPO MESSAGE DELIVERY DETAILS:
1 N  N10628 ..NO NPF MESSAGE RESTRICTIONS.
F1=Help   F2=Split   F3=Exit   F4=Return   F5=Find   F6=Refresh
F7=Backward F8=Forward F9=Swap   F11=Right

```

Modify a Printer Entry

You can modify a print request to change where and how it is to be printed.

To modify a print request

1. Enter **M** next to the required print request in the PSM: Output Queue.

The PSM : Print Request panel appears.

The PSM : Print Request Panel provides all details about the print request. You can alter some of the fields on the panel.

Confirm Printing

When you send a print request to a printer, the PSM : Confirm Printer panel appears. This panel is used to confirm the printer name, the number of copies, and the hold and keep settings that you require for your print request. The fields displayed on the panel are set to the values you used last.

To change any of these fields, overwrite them with the required information, and press F6 (Confirm).

The new information is used to print your request.

Note: For more information about the fields displayed on this panel, press F1 (Help).

Select the Printer

If you do not know what printers are available to send your print request to, you can display a list of active printers.

To select the printer

1. Enter a question mark (?) in the Printer Name field on the PSM: Print Request panel.

The list of active printers appears.

Note: If the list is longer than a full page, use F8 (Forward) and F7 (Backward) to scroll through the list.

2. Enter the selection code at the prompt.

The printer is selected.

Appendix A: File Types and Attributes

This section contains the following topics:

[Supported File Attributes](#) (see page 102)

[Supplemental Data Set Organization](#) (see page 103)

Supported File Attributes

The following table shows the types of attributes supported for the input and output data sets and the possibilities that exist for reblocking during transmission. Only the generic type of attribute is shown (fixed, variable, or undefined) but data sets may also be blocked and contain printer control characters.

Note: Special considerations apply when transmitting from, or into, Generation Data Groups (GDGs).

		OUTPUT			
Record Format		F i x e d	V a r i a b l e	V a r i a b l e S P a n	U n d e f i n e d
I N P U T	Fixed	R	R	I	I
	Variable	I	R	I	I
	Variable Span	I	I	I	I
	Undefined	I	I	I	I

I

File format is supported only for an image-copy transmission or if the input data set attributes are identical to the output. The output data set attributes are overridden with those of the input data set. IMAGE-COPY is an option on a transmission definition.

R

File format is supported. If required and not an IMAGE-COPY transmission, the input file will, if possible, be reblocked on receipt to match the attributes of the output file or those specified in the transmission definition.

Supplemental Data Set Organization

The following table shows the data set organizations supported for the input and output data sets. Where a particular organization is valid for transmission, file attribute checking is also performed as shown in [Supported File Attributes](#) (see page 102).

Data Set Organization		OUTPUT				
		S e q u e n t i a l	D i r e c t	P a r t i t i o n e d (1,2)	V S A M	S Y S O U T (4)
I N P U T	Sequential	Y	Y	Y	N	Y
	Direct	Y	Y	Y	N	Y
	Partitioned (1)	Y	Y	Y	N	N
	VSAM (3)	N	N	N	N	N
	SYSIN (4)	Y	Y	Y	N	Y

Y

File format is supported.

N

File format is not supported.

Notes:

- Specify only one member of the partitioned data set if you are defining transmission definitions through FTS Transmission Definition full-screen panels. Multiple members can be transmitted using the FTS External Interface facilities, or by unloading the members into a sequential data set using a utility such as IEBCOPY and transmitting that data set. Members that contain user note lists within the directory (for example, load modules) are supported only as unloaded sequential data sets. Restart is not supported if the input or output data set is a member of a partitioned data set.
- Must not be an IMAGE-COPY transmission unless the input and output data set attributes are identical (that is, any existing output data set attributes cannot be changed).
- Can be supported by first creating a sequential data set using a utility such as IDCAMS, and transmitting that data set.
- Restart is not supported if the input data set is SYSIN or the output data set is SYSOUT.

Appendix B: Generation Data Groups

This section contains the following topics:

[GDGs Support](#) (see page 105)

[Cold Start a Request Using GDGs](#) (see page 107)

[Potential Errors with GDGs](#) (see page 108)

[Collective Concatenation of Data Sets in a GDG](#) (see page 109)

GDGs Support

CA SOLVE:FTS supports GDG transmission and the receipt of data sets into GDGs. The GDGs usage can significantly extend the flexibility associated with transmitting data at regular intervals, perhaps with the use of timer commands.

GDG usage is indicated when the FROM data set name or the TO data set name on the transmission definition panel is specified in *one* of the following formats:

A.B(0)

A.B(+*n*)

A.B(-*n*)

The absolute generation number, as represented by (+*n*) or (-*n*), is determined during the first allocation of the data set before the commencement of transmission. This determination is made when the transmission request is first scheduled rather than when the transmission definition or transmission request is entered. There may be a delay between entering a transmission request and its scheduling, pending the availability of the link to the receiving host and a suitable transmission initiator to perform the transmission.

When the data set represented by the specified relative generation number (as at the start of the transmission) is identified, CA SOLVE:FTS is not affected by subsequent changes caused by new generations being added to the GDG.

GDS Support Example

Consider the case where a GDG with a data set name of GDG.IN is transmitted to a sequential data set with a data set name of SEQ.OUT. The generation data group was defined to support a maximum of four entries. Entries in excess of this number cause the operating system to delete the oldest entry. Currently three generations exist. They are GDG.IN.G0003V00, GDG.IN.00002V00, and GDG.IN.00001V00.

Note: The fourth generation GDG.IN.00004V00 is not yet created.

The current relative generations are:

Absolute Generation	Relative Generation
GDG.IN.00003V00	GDG.IN(0)
GDG.IN.G0002V00	GDG.IN(-1)
GDG.IN.00001V00	GDG.IN(-2)

For example, assume that we want to transmit the latest generation, GDG.IN(0), to the data set SEQ.OUT.

The transmission definition is defined with the FROM data set as GDG.IN(0) and the TO data set as SEQ.OUT, the transmission request is entered and the request scheduled.

When scheduled, CA SOLVE:FTS analyzes the FROM data set name and determines that a GDG is transmitted. It then determines the absolute generation number at that time. In our example, this is the data set GDG.IN.00003V00. Having determined the actual data set name to use, it remembers this name and proceeds with the transmission.

Any displays of the transmission request from this point gives the absolute data set name being used (in our case GDG.IN.G0003V00) rather than the relative generation number as entered in the original transmission definition. The original transmission definition has not been changed and is still GDG.IN(0). This technique must be used to ensure that CA SOLVE:FTS is not affected by subsequent changes to the GDG by the addition of extra data sets during the transmission.

Assume that our request commences transmission, but, before completion, the transmission is interrupted by a link failure. The link failure may prevent the restart of the transmission until the fault on the link is corrected. During this time, a new data set is added to the GDG.

The status of the GDG is now:

Absolute Generation	Relative Generation
GDG.IN.G0004V00	GDG.IN(0)
GDG.IN.G0003V00	GDG.IN(-1)
GDG.IN.00002V00	GDG.IN(-2)
GDG.IN.G0001V00	GDG.IN(-3)

The link fault is corrected, and CA SOLVE:FTS can restart the request. If the request restarted on the data set specified in the original transmission definition, (GDG.IN(0)) the incorrect data set GDG.IN.G0004V00 would be selected and the transmission would either fail with I/O errors or incorrect data would be transmitted.

CA SOLVE:FTS protects against this situation by resolving the data set name before starting the transmission and the checkpointing this resolved name. Thus, transmission is restarted correctly on the data set GDG.IN.G0003V00.

The same applies at the receiver, if the receiving data set is a GDG that was specified as a relative generation number.

You can also enter the data set names as absolute generation numbers, for example, MY.GDG.G0123V00.

Cold Start a Request Using GDGs

As with any other transmission request, you can interrupt the transmission of a GDG and, if required, request a cold start be performed. A cold start implies that transmission of the data set be restarted from the beginning of the data set rather than any checkpoint that may be valid.

In such a case, the previously resolved absolute generation number is again used, and any changes to the GDG that occurred since the transmission was originally started are ignored.

If you want the current status of the GDG to apply and the relative generation number to be resolved again to its current absolute number, then you must delete the request and re-enter it.

Potential Errors with GDGs

The number of entries in the GDG can be an important factor. Depending on how the GDG was originally established, the number of entries in a GDG can determine the point at which either the oldest generation or all of the generations are deleted by the operating system.

Assume the addition of yet another data set to the GDG in our earlier example. The earliest generation, GDG.IN(-3), which is actually data set GDG.IN.G0001V00, would be deleted and the new generation added, resulting in the following:

Absolute Generation	Relative Generation
GDG.IN.G0005V00	GDG.IN(0)
GDG.IN.G0004V00	GDG.IN(-1)
GDG.IN.G0003V00	GDG.IN(-2)
GDG.IN.00002V00	GDG.IN(-3)

CA SOLVE:FTS plays no part in this deletion. It is purely a function of the way the operating system supports GDGs.

Because of this fact, the number of entries and possible auto-deletion of required data sets may impact the ability of a failed transmission to be restarted. This is always the case if the data set is deleted during transmission or before a restart can take place.

Important! The installation must therefore ensure that sufficient entries exist in the GDG to allow for possible interrupted transmissions and required restarts.

Collective Concatenation of Data Sets in a GDG

You can refer collectively to all of the data sets in a GDG as a single data set for input by omitting the generation number when you define the transmission definition. This causes all data sets in the GDG to be treated as one sequential data set, beginning with the current relative generation 0 and ending with the oldest data set (-n) in the GDG.

Use of collective concatenation makes it possible to cater to accumulate data before transmission and eliminates the need to know the exact number of generations that need to be transmitted. In our example, entry of GDG.IN would imply all data sets that currently exist within the GDG.

Restart of transmissions that refer collectively to all data sets in a GDG is not supported and a cold start is performed. The data sets that exist in the GDG at the time of the cold start are used. In such cases there is no resolution or checkpointing of the individual data set names in the GDG when the transmission request is scheduled.

Attempts to collectively reference (for instance, no generation number is specified) a GDG as an output data set are rejected. For *new* data sets, this results in dynamic allocation errors because the catalog index structure is incorrect. For existing data sets (OLD or SHR), the transmission fails with an appropriate completion code, because allowing it to proceed would result in the entire transmission being written to the current relative generation 0 data set. Thus, for output to a GDG, the TO data set name must always specify either a relative or absolute generation number.

An alternative to collective concatenation of GDGs is using a system utility to copy the entire GDG to a sequential data set before transmission and performing the transmission from that data set.

Note: While Generation Data Groups offer great potential for improving operational efficiency, you must consider the procedures established for their use, keeping the above points in mind.

Index

\$

\$USTSFTS • 74, 84, 89

&

&FTS variables
 subcommand environment, in • 74

A

access authorization exit • 76
action lists • 21
activity logs
 record additional information • 60
activity reporting • 16
automatic activation • 13

B

background mode • 74
batch jobs • 71, 74

C

commands
 timer • 71
commands, SHOW
 SHOW FTS • 15, 70
commands, specific
 FTSINIT • 70, 71
 FTSMOD • 70
 LOCATE • 23
 PROFILE • 69
 SPLIT • 25
 SWAP • 25
 TRANSMIT • 69, 71
contacting technical support • 3
customer support, contacting • 3

D

data sets
 copying • 94
 editing • 94
 PDS • 96
 prefixing • 76
 reblocking • 101
 receiving into GDGs • 105

 temporary • 96
 transmitting • 71
 VSAM • 94, 96
DCB attributes • 76
definitions
 copy • 89
 create • 76, 89
 delete • 82, 86
directory block information • 76

E

editing
 data sets • 94
 restrictions • 96
error recovery • 15, 32
external interface • 84

F

failed requests, error codes • 16
file transfer
 direct • 96
 staged • 96
 transmission status • 15
Find function • 23
FROM command • 76

G

Generation Data Groups (GDGs) • 101

H

help, online
 NMCMD command • 75
 overview • 23
horizontal scrolling • 22

I

image copy • 76
inbound transmissions, concurrent • 37
initiators
 example definition • 37
 maximum • 37
 modifying • 70
 starting • 71
 stopping • 71

invalid call • 86

ISPF • 94

- dialog interface • 93

- options • 94

- overview • 93

- panel • 89

L

list types • 21

listing destinations • 36

local host • 93

LOCATE command • 23

locating records • 23

M

messages

- dialog, in • 94

- help for • 24

- returned to TSO • 86

- subcommand environment, in • 74

- suppression using PROFILE command • 69

multiple select lists • 21

N

navigation

- splitting screens • 25

- swapping screens • 25

- toggling between windows • 25

- working in two windows • 25

NCL interface • 73

NCL procedures • 83

- interface with TSO • 73

- restrictions • 74

- TSO • 84

NMCMD command processor

- help, command • 75

NMFTSID • 94

numbered lists • 21

O

online help

- messages • 24

- NMCMD command • 75

- overview • 23

Operator Console Services (OCS) • 11

P

partitioned data set (PDS)

- dialog, in • 94

- FTSDEF subcommand • 76

- load library members • 96

- staged transmissions • 96

private definition • 86, 89

R

rejected requests • 37

remote host • 93

return codes • 86, 89

S

screens

- splitting • 25

- swapping • 25

scrolling • 22

security

- access masks • 20

- privileges • 20

single select lists • 21

space allocation

- directory • 76

- primary • 76

- secondary • 76

SPLIT command • 25

staging data set • 96

support, contacting • 3

SWAP command • 25

swapping screens • 25

system definition • 76, 89

T

tailoring • 94

technical support, contacting • 3

tip of the day • 23

TO command • 76

toggling between windows • 25

transmission definitions

- adding • 11

- deleting • 11, 82

- maintaining • 29

- modifying • 11

- storage • 76

- transmitting • 84

transmission initiator

- define • 13
- number of transmissions • 13
- processing allocation of classes • 13
- processing priority • 13
- processing requests • 13
- transmission requests
 - altering • 12
 - automatic rescheduling • 15
 - deleting • 12
 - failed • 15
 - failed, error codes • 16
 - failures • 74
 - issuing • 12, 32
 - listing • 32
 - modifying • 70
 - monitoring • 15
 - name • 12
 - processing • 12
 - resetting failed • 15
 - restarting • 15
 - scheduling • 14
 - status • 12, 74
- transmissions
 - displaying • 15
 - enabling and disabling • 71
 - listing • 15
 - status • 70
 - using NCL procedures • 70
- transmitted data
 - compression • 13, 17
 - routes through the network • 13, 16
- TSO
 - command processor • 84, 89, 92
 - PUTLINE macros • 82

U

- unit information • 76
- user services • 84

V

- vertical scrolling • 22
- VFS • 11
 - database • 11, 82, 84, 86
 - file error • 89
- VSAM data sets • 94, 96

W

- working in two windows • 25