Connect:Express™
z/OS

PeSIT User Fields Guide

Version 4.2.3

Connect:Express z/OS PeSIT User Fields Guide
Version 4.2.3
First Edition

This documentation was prepared to assist licensed users of the Connect:Express system ("Sterling Commerce Software"). The Sterling Commerce Software, the related documentation and the information and know-how it contains, is proprietary and confidential and constitutes valuable trade secrets of Sterling Commerce, Inc., its affiliated companies or its or their licensors (collectively "Sterling Commerce"), and may not be used for any unauthorized purpose or disclosed to others without the prior written permission of Sterling Commerce. The Sterling Commerce Software and the information and know-how it contains have been provided pursuant to a license agreement which contains prohibitions against and/or restrictions on its copying, modification and use. Duplication, in whole or in part, if and when permitted, shall bear this notice and the Sterling Commerce, Inc. copyright legend.

Where any of the Sterling Commerce Software or Third Party Software is used, duplicated or disclosed by or to the United States government or a government contractor or subcontractor, it is provided with RESTRICTED RIGHTS as defined in Title 48 CFR 52.227-19 and is subject to the following: Title 48 CFR 2.101, 12.212, 52.227-19, 227.7201 through 227.7202-4, FAR 52.227-14(g)(2)(6/87), and FAR 52.227-19(c)(2) and (6/87), and where applicable, the customary Sterling Commerce license, as described in Title 48 CFR 227-7202-3 with respect to commercial software and commercial software documentation including DFAR 252.227-7013(c) (1), 252.227-7015(b) and (2), DFAR 252.227-7015(b)(6/95), DFAR 227.7202-3(a), all as applicable.

The Sterling Commerce Software and the related documentation are licensed either "AS IS" or with a limited warranty, as described in the Sterling Commerce license agreement. Other than any limited warranties provided, NO OTHER WARRANTY IS EXPRESSED AND NONE SHALL BE IMPLIED, INCLUDING THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR USE OR FOR A PARTICULAR PURPOSE. The applicable Sterling Commerce entity reserves the right to revise this publication from time to time and to make changes in the content hereof without the obligation to notify any person or entity of such revisions or changes.

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

Printed in the United States of America.

Copyright © 2003, 2009. Sterling Commerce, Inc. All rights reserved.

Connect:Express is a registered trademark of Sterling Commerce. All Third Party Software names are trademarks or registered trademarks of their respective companies. All other brand or product names are trademarks or registered trademarks of their respective companies.

CXPESITG703

Contents

Appendix A   Keywords

Appendix B   Using Environment Variables

Preface

The *Connect:Express z/OS PeSIT User Fields Guide* is for programmers and network operations staff who automate and organize file transfer operations with Connect:Express z/OS.

Chapter Overview

The *Connect:Express z/OS PeSIT User Fields Guide* is organized into the following chapters and appendix:

Chapter/Appendix	Description
1 Introduction	This chapter provides an overview of the PeSIT user fields Pi37 and Pi99, and describes how information can be transferred with a non-Connect:Express Partner.
2 Requester	This chapter describes the use of Pi37 and Pi99 when a requester transmits or receives a file. The Pi99 sent by the requester contains complementary information for executing the transfer, and the Pi99 sent by the server contains complementary information associated with the server response.
3 Server	This chapter describes the use of Pi37 and Pi99 when a server transmits or receives file. The Pi99 sent by the requester contains complementary information for executing the transfer, and the Pi99 sent by the server contains complementary information associated with the server response.
A Keywords	This appendix lists the keywords that you can use in PeSIT user fields with a non-Connect:Express partner.

Connect:Express Documentation

Connect:Express documentation consists of the following manuals:

❖ The *Connect:Express z/OS 4.2.3 Release Notes* lists maintenance updates and any important notes.

❖ The *Connect:Express z/OS 4.2.3 Installation Guide* describes the planning and installation of Connect:Express.

❖ The *Connect:Express z/OS 4.2.3 User Guide* includes general information on using the TSO/ISPF interface, and serves as a reference of user and environment commands.

❖ The *Connect:Express z/OS 4.2.3 Utilities Guide* describes the optional Utilities package that you can integrate with Connect:Express.

❖ The *Connect:Express z/OS 4.2.3 FTP Guide* provides you with the information that you need to use Connect:Express with the FTP protocol.

❖ The *Connect:Express z/OS 4.2.3 Administration Guide* provides detailed information about transfer operations for system administrators and other advanced users of Connect:Express.

❖ The *Connect:Express z/OS 4.2.3 Options Guide* provides information about the CICS, IMS, and RJE interfaces available for Connect:Express.

❖ The *Connect:Express z/OS 4.2.3 PeSIT User Fields Guide* describes how you can exchange the PeSIT Pi37 and Pi99 fields with any PeSIT software.

❖ The *Connect:Express HTTP Option Implementation Guide* provides you with the information that you need to implement HTTP access to the Connect:Express z/OS repository.

❖ The *Connect:Express z/OS 4.2.3 Etebac3 User Guide* provides you with the information that you need to use Connect:Express with the Etebac3 protocol.

❖ The *Connect:Express z/OS 4.2.3 SSL Guide* includes general information on implementing secured file transfers.

❖ The *Connect:Express z/OS 4.2.3 Sysplex Supervision Guide* includes general information on implementing a group of Connect:Express Plex managers under control of a Connect:Express Plex supervisor.

Getting Support for Sterling Commerce Products

Sterling Commerce provides intuitive technical products and superior Help and documentation to enable you to work independently. However, if you have a technical question regarding a Sterling Commerce product, use the Sterling Commerce Customer Support Web site.

The Sterling Commerce Customer Support Web site at *www.sterlingcommerce.com* is the doorway to Web support, information, and tools. This Web site contains several informative links, including a solutions database, an issue tracking system, fix information, documentation, workshop information, contact information, sunset and retirement schedules, and ordering information. Refer to the Customer Support Reference Guide at *www.sterlingcommerce.com/customer/tech_support.html* for specific information on getting support for Sterling Commerce products.

Conventions Used in This Guide

The *Connect:Express z/OS PeSIT User Fields Guide* uses certain notational conventions. This section describes the conventions used in this guide.

Convention	Description
UPPERCASE LETTERS	Uppercase letters in the command format indicate that you type in information as shown.
Lowercase letters	Lowercase letters or words in commands or syntax boxes require substitution by the user. For example, index1.index2.PARMLIB indicates that you must provide the first and second indexes of the string. "PARMLIB" is mandatory.
Bold Letters	Bold print in syntax boxes indicates Connect:Express commands and required parameters. For example, PLEX=N indicates that the parameter PLEX must be set to N.
Underlined Letters	Underlining indicates default values for parameters and subparameters. For example, PLEX=Y|N specifies that the default for PLEX is N.

Convention	Description
Vertical Bars (|)	Vertical bars indicate that you can supply one of a series of values separated by the vertical bars. For example RUN=H|C specifies that H or C is valid.
`Monospaced characters` (characters of equal width)	`Monospaced characters` represent information for screens, commands, Processes, and reports.
Punctuation	Code all commas and parentheses as they appear.
£ or #	The Pound character (£) and the hash character (#) are equivalent.

Chapter 1

Introduction

This chapter provides an overview of the PeSIT user fields Pi37 and Pi99, and describes how information can be transferred with a non-Connect:Express Partner.

PeSIT User Fields and Connect:Express

In earlier versions of Connect:Express, the PeSIT user fields could only be used between Connect:Express Partners, and the information transferred in these fields was unique to Connect:Express. With Connect:Express z/OS version 4.1.0, the use of the PeSIT user fields Pi37 and Pi99 was expanded, and new keywords were added for these fields. You can now use the PeSIT user fields with non-Connect:Express Partners, and the information transferred in these fields can be understood by any PeSIT user. To use this new functionality, you must designate a Partner as type Other in the Partner definition.

Note: A non-Connect:Express Partner does not mean that the remote monitor is not a Connect:Express monitor, only that the Connect:Express proprietary protocol for Pi37 and Pi99 is not used.

Pi37 and Pi99

The Pi37 and Pi99 fields (Pi = protocol identifier) are elements of the PeSIT protocol that enable you to exchange free-form information between Partners during a file transfer. This ability to manage information outside the protocol can be used by applications that manage transfers when transmitting or receiving. The Pi37 and Pi99 fields are transmitted during the file selection phase.

The maximum size of data that can be transmitted in the Pi37 field is 80 characters. For the Pi99 field, the maximum size varies depending on the version of PeSIT that you are using. The capacity of Pi99 ranges from 64 characters with PeSIT D to 254 characters with PeSIT E.

By definition, the Pi37 field corresponds to the label of the file, or the physical file name on the transmitter's system. Pi37 is driven by the PeSIT File transfer Protocol Data Units (fpdus) *Create* and *Ack-Select*. Therefore, it can only be sent by the transmitter, either the requester or server. This field is an optional field.

Pi99 is also an optional field. With PeSIT E, it is driven by the PeSIT fpdus *Create*, *Select*, *Ack-Create* and *Ack-Select*. Therefore, it can be sent by the transmitter or receiver, requester or server. With PeSIT D, it is driven by the fpdus *Create, Select,* and *Ack-Select,* and cannot be sent by the server receiving the file.

The two parameters are used by different PeSIT products depending on the mode, either proprietary or open.

Depending on the product with which you communicate (CFT or INTERPEL, XFB for example), the use of a parameter may be required. Connect:Express sends parameters with default values, but in certain cases, they can be entered at the time of the transfer request or changed by a user exit. They are handled by Connect:Express at the monitor level and at the transfer process level, internally or with keywords.

The values in the two Pi fields depend on if the partner is defined as a Connect:Express partner (type=TOM) or a non-Connect:Express partner (type=Other) in the Partner directory. If both Partners are Connect:Express Partners, Pi37 and Pi99 are filled in based on the proprietary mode that already exists in Connect:Express MVS 2.2.2. This functionality remains unchanged.

This document describes the use of Pi37 and Pi99 by Connect:Express z/OS in the case where the remote monitor is a non-Connect:Express monitor or where a Connect:Express monitor has defined the local Partner as a type Other. The functionality of the local Partner is described in the following four cases:

❖   Transmitting Requester

❖   Receiving Requester

❖   Transmitting Server

❖   Receiving Server

Sending Pi37 and Pi99 with Connect:Express

Several methods are available to place information in the Pi37 and Pi99 fields. You can save this information in the symbolic file definition, pass it in the parameters of a transfer request, or ask the general exit to provide the information. Lower case characters are supported and it is possible to include keyword variables that are resolved at the time the transfer is executed. See Appendix A, *Keywords*, for a list of keywords that you can use with Connect:Express.

Note:	If you include vanables in these fields, use upper case characters. (example: &REQNUMB)

Parameter of the Symbolic File

The field Remote DSN/Pi99 enables you to save a text string of 44 characters that Connect:Express will place in the Pi99 field. This value can be replaced by the parameter in the transfer request.

Note:	If the Pi99 is equal to the text string "PI37", this implies that the Pi99 will contain the same value as that transmitted in the Pi37.

Parameters of the Transfer Request

You can implement these two parameters in a transfer request issued by an operator, a batch utility, or the program interface. The field RDSN/Pi99 enables you to send the Pi99 and use the Api field either to send the Pi99 or to send the Pi37.

Note:	If the Pi99 is equal to the text string "PI37", this implies that the Pi99 will contain the same value as that transmitted in the Pi37.

Transmitting the Pi99

The Rdsn/Pi99 field enables you to save a text string of 44 characters that Connect:Express will place in the Pi99. This value replaces the default value saved in the file definition in the files directory, and can be modified by the general exit. The text string can contain symbolic variables resolved locally or at the remote partner's site.

In the following example, the resulting Pi99 will contain the request number of the transmitter, followed by the request number of the receiver.

```
     Rdsn/Pi99 .. ---> &REQNUMB -  &&REQNUMB
```

Use the syntax shown below to save a text string of 80 characters in the Api field. Connect:Express will place this string in the Pi99 field. The prefix P tells Connect:Express to place the text string that follows in the Pi99 field.

```
     Api ....  ---> P:Pi99 of 80 characters
```

This value replaces the default value saved in the file definition in the files directory, and can be modified by the general exit. The text string can contain symbolic variables resolved at the remote partner's location.

Transmitting the Pi37

Use the syntax shown below to save a text string of 80 characters in the Api field. Connect:Express will place this string in the Pi37 field. The prefix U tells Connect:Express to place the text string that follows in the Pi37 field.

```
     Api ....  ---> U: &REQNUMB -  &&REQNUMB
```

This value can be modified by the general exit. The text string can contain symbolic variables resolved at the remote partner's location. In the preceding example, the resulting Pi37 will contain the request number of the transmitter followed by the request number of the receiver.

Using the General Exit

The general exit provides several keywords that you can send in the Pi99 and Pi37 fields. The parameters of these keywords can contain symbolic variables that the general exit resolves before replacing the contents of the PeSIT fields.

Transmitting the Pi99

Two keywords enable you to send a Pi99 of 254 characters. The keyword $PI99$ receives the Pi99 field to send. The keyword $PI99RD$ receives a file name containing the Pi99 field to send.

The value of Pi99 processed by the exit replaces the value associated with the transfer request.

The keyword $PI99RD$ enables an application to compose a Pi99 of 254 characters and pass it dynamically before executing the transfer.

Transmitting the Pi37

The keyword $PI37$ receives the Pi37 field (up to 80 characters) to send in a parameter. The value of Pi37 processed by the exit replaces the value associated with the transfer request.

Reception of Pi37 and Pi99 by Connect:Express

Connect:Express receives and processes the Pi37 and Pi99 fields at different points, corresponding to the phases of a file transfer. Depending on the configuration, Connect:Express processes these fields using the maximum field length,or only the first 44 characters.

Physical Name of the File to Transfer

The symbolic variables &PI99 and &EXTLAB enable you to integrate the first characters of each of these fields in the physical file name, as long as the result produced corresponds to a valid structure of the physical file name on MVS. If these fields contain symbolic variables, Connect:Express resolves them before resolving the variables of the fields themselves.

The following example shows how to construct a physical file name equal to the local request number from the Pi99.

```
DSN LOCAL ............. ===> LOCAL.APPLI.&PI99
Pi99 received :  &REQNUMB  - Local request number = 00000012
DSN final = LOCAL.APPLI.A0000012
```

Start of Transfer Commands

Connect:Express can only pass one of the two parameters (either Pi99 or Pi37), up to a maximum of 44 characters, to a start of transfer command. In the case of a transmission, Connect:Express can only receive the Pi99. But in the case of a reception, it receives both parameters. For this reason, if the Pi99 is empty, Connect:Express replaces it with the Pi37. Therefore, to be able to process the Pi37 in the start of transfer command, the Pi99 received must be empty.

If the field contains symbolic variables, Connect:Express resolves them before resolving the variable in the field itself.

Receiving the Pi99

Chapters 2 and 3 describe the reception of the Pi99 when the protocol used is PeSIT D or PeSIT E. Since Connect:Express receives the Pi99, it resolves the symbolic variables and passes them in parameters to the start of transfer command using the keyword &Pi99. The following screen shows an example.

```
_ START COMMAND ....... ===> &PI99
```

The preceding example indicates that the start of transfer command is located in the Pi99 field. This can corresponds to the configuration at the Connect:Express remote site.

```
Api ....  ---> P:SE '&FILENAM in Reception',USER=(USR)
```

Receiving the Pi37

The Pi37 is always received by the destination of the file. When Connect:Express receives the Pi37, it resolves the symbolic variables and passes them in parameters to the start of transfer command using the keyword &EXTLAB, as long as the parameter Pi99 received is empty. The following screen shows an example.

```
_ START COMMAND ....... ===> &EXTLAB
```

The preceding example indicates that the start of transfer command is located in the Pi37 field. This can correspond to the configuration at the Connect:Express remote site.

```
Api ....  ---> U:SE '&FILENAM in Reception',USER=(USR)
```

Using the General Exit

Depending on the context, the general exit can receive the two parameters, or only one. Different symbolic variables and keywords are available. The general exit enables you to process the two parameters at the same time, using the entire field length.

Receiving the Pi99

The symbolic variable &PI99 can receive 254 characters and be processed by the general exit.

```
$CMD$    SEND '&PI99',USER=(USER),LOGON
```

The keyword $PI99WD$ enables you to save the 254 characters of the Pi99 received in a file for processing later by an application.

```
$PI99WD$ INDEX1.INDEX2.&REQNUMB.P99RECU
```

Receiving the Pi37

The symbolic variable &PI37 can receive 80 characters of the parameter and be used in a Process executed by the exit.

```
$CMD$    SEND '&PI37',USER=(USER),LOGON
```

End of Transfer Commands

Connect:Express can only pass one of the two parameters (either Pi99 or Pi37), up to 44 characters, to an end of transfer command. In the case of a transmission, Connect:Express can only receive the Pi99. But in the case of a reception, it receives both parameters. For this reason, if the Pi99 received is empty, Connect:Express replaces it with the Pi37. In other words, to be able to process the Pi37 in the end of transfer command, the Pi99 received must be empty.

The field Pi99 or Pi37 processed is the same as that of the start of transfer commands. If the field contains symbolic variables, Connect:Express resolves them before resolving the field itself.

Note: You must differentiatebetween the Pi99 transmitted and the Pi99 received. These two fields are displayed in the Journal, in the fields USDF.

```
USDF <-:
USDF ->:
```

In the case of a reception, the field USDF <- displays the Pi99 or the Pi37, depending on if the Pi99 received was empty or not.

Substitution of Variables by Connect:Express

The text strings provided in the PeSIT fields can contain symbolic variables in the form, &VAR, or environment variables in the form, <var>, that Connect:Express resolves at the time of the transfer, either before sending them to the partner or after receiving them.

The availabe symbolic variables are listed in Appendix A. The use of environment variables is described in Appendix B. Using these variables, you can dynamically construct the user fields based on the context (request number, date and time of the transfer for example). The syntax '&&VAR' gives you the possibility of sending the variable &VAR in the PeSITuser field. In this case, the variable is resolved by the remote partner.

Substitution of Symbolic Variables in the Pi37 Transmitted to a Partner

You can include symbolic variables in the entry field API U of the Pi37 of the transfer request, or with the keyword $PI37$ of the general exit. Connect:Express substitutes the current values at the time that the Pi37 is created and before the transmission of the PeSIT fpdu. See Appendix A for a list of keywords. If the substitution exceeds the capacity of the Pi37 field (80 characters), the text string is truncated.

Depending on the context, the list of authorized variables is different. Refer to chapters 2 and 3 for more information.

Substitution of Symbolic Variables in the Pi37 Received from a Partner

The Pi37 sent by the partner can contain symbolic variables. Connect:Express substitutes the current local values before using the result in the variables &EXTLAB or &PI37. See Appendix A for a list of keywords. If the substitution exceeds the capacity of the Pi37, the text string is truncated.

Depending on the context, the list of authorized variables is different. Refer to chapters 2 and 3 for more information.

Substitution of Variables in the Pi99 Transmitted to a Partner

You can include symbolic variables and environment variables in the entry field *Remote dsn/Pi99* of the file definition or the transfer request, or with the keyword $PI99$ of the general exit. Connect:Express substitutes the current values at the time that the Pi99 is created and before the transmission of the PeSIT fpdu. See Appendix A for a list of keywords, and Appendix B for using environment variables. If the substitution exceeds the capacity of the Pi99 field (254 characters), the text string is truncated.

Depending on the context, the list of authorized variables is different. Refer to chapters 2 and 3 for more information.

Substitution ofVariables in the Pi99 Received from a Partner

The Pi99 sent by the partner can contain symbolic variables and environment variables . Connect:Express substitutes the current local values before using the result in the variable &PI99. See Appendix A for a list of keywords, and Appendix B for using environment variables. If the substitution exceeds the capacity of the Pi99, the text string is truncated.

Depending on the context, the list of authorized variables is different. Refer to chapters 2 and 3 for more information.

Requester

This chapter describes the use of Pi37 and Pi99 when a requester transmits or receives a file. The Pi99 sent by the requester contains complementary information for executing the transfer, and the Pi99 sent by the server contains complementary information associated with the server response.

Transmitting Requester

When a requester is the transmitter, the requester issues a request to send a local file to a remote partner. The exchange of PeSIT fpdus during the selection phase is shown in the following diagram:
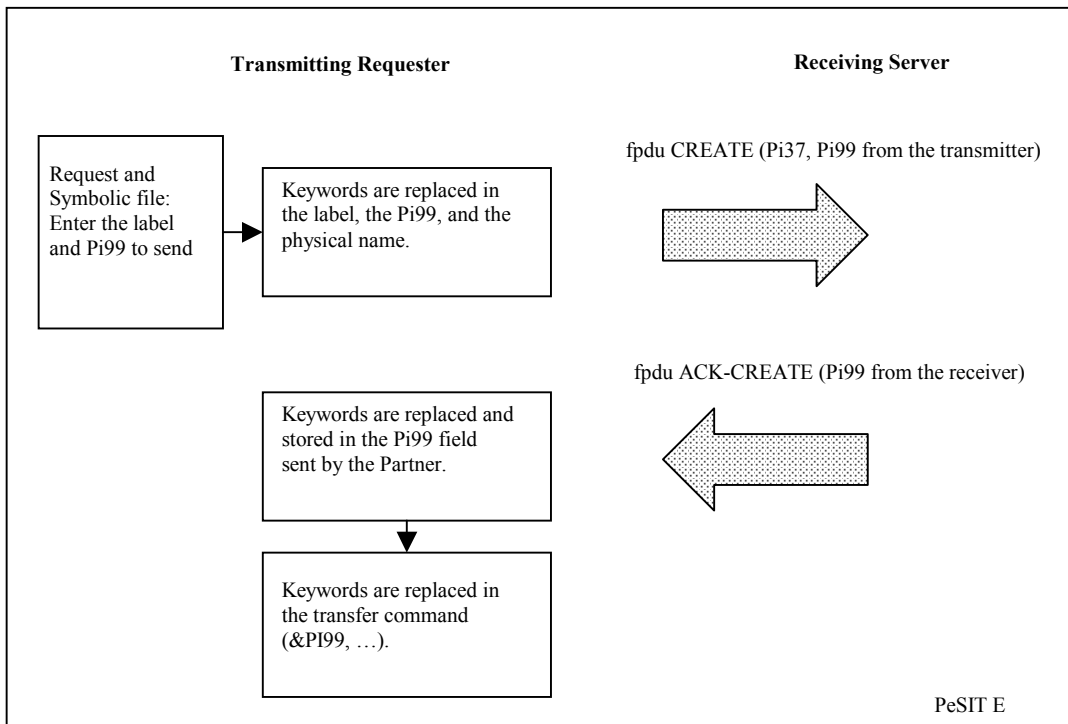
Note: With PeSIT D, the receiving server does not send Pi99.

Transmitting Pi37

You can enter a value in the Pi37 field at the time of the transfer request in any of the following ways:

❖ In the API field, U: format when a request is entered with the user interface TSO/ISPF 4.1 Request Extension screen.

❖ In the parameter API=U: of the batch utilities P1B2PREQ and P1B2PRQ3.

❖ In the parameters API=U: of the batch utility P1B2PREQ.

❖ Using the $PI37$ keyword of the Generalized exit L1GFIUE1.

By default, if the Label field is not entered with the request, then the complete file name of the transmitted file is sent in Pi37.

Some keywords can be inserted in the Label field. They are replaced by the values that are valid at the time Pi37 is generated, and before the transmission of the fpdu Create. See the Appendix for more information about keywords.

For example, if the date and time of the request are 20020501 and 195312, and the Partner is PART01, the entry &PARTNID_&REQDATE_&REQTIME.txt in the Pi37 field results in the following:

```
PART01_20020501_195312.txt
```

If the information is more than 80 characters, the string is truncated. The following keywords can be used with Pi37:

&PARTNID, &FILENAM, &ORG, &DST, &REQNUMB, &REQDAT, &REQTIM, &DIR, &UID, &DSN

Transmitting Pi99

You can enter information in Pi99 in any of the following ways when transmitting a file:

❖ In the symbolic file definition used by the transfer request. If you are using the user interface TSO/ISPF 1.2, enter this in the REMOTE DSN/PI99 field.

❖ In the 44-character REMOTE DSN/PI99 field when a request is entered with the user interface TSO/ISPF 4.1, Request Extension screen.

❖ In the parameter RDS= of the batch utilities P1B2PRQ2, P1B2PRQ3 and P1B2PREQ.

❖ In the 82-character API field, P: format when a request is entered with the user interface TSO/ISPF 4.1 Request Extension screen.

❖ In the parameter API=P: of the batch utilities P1B2PREQ and P1B2PRQ3.

❖ In the parameter API=P: of the batch utility P1B2PREQ.

❖ Using the $PI99$ keyword of the Generalized exit L1GFIUE1.

❖ Using the $PI99RD$ keyword of the Generalized exit L1GFIUE1.

> Note: The REMOTE DSN and API fields cannot be used at the same time. Use The REMOTE DSN field if you need 44 characters, and the API field if you need 80 charcaters. Use the Generalized exit L1GFIUE1 (keywords $PI99$ and $PI99RD$) if you need 254 characters.

The value is placed in the Pi99 field that you want to send. If the value of Pi99 is not entered in the initial transfer request, Connect:Express enters the value found in the associated symbolic file definition. If no value is found, the Pi99 field is not transmitted. The Pi99 provided by the begining of transfer user exit replaces any string provided previously.

You can insert keywords in the Value field of Pi99. They are replaced by the values that are valid at the time Pi99 is generated and before the transmission of the fpdu Create. See the Appendix for more information about keywords.

If the information exceeds the capacity of Pi99, the string is truncated. You can use the following variables with Pi99:

&PARTNID, &FILENAM, &ORG, &DST, &REQNUMB, &REQDAT, &REQTIM, &DIR, &DSN, &UID

Receiving Pi99 (PeSIT E)

You can use keywords in the Pi99 received from a Partner. These keywords can then be replaced by valid values on the local system. See the Appendix for more information about keywords.

If the information exceeds the capacity of the Pi99 field, the string is truncated. You can use the following keywords with the Pi99 field when receiving it from a Partner:

&PARTNID, &FILENAM, &ORG, &DST, &REQNUMB, &REQDAT, &REQTIM, &DIR, &DSN, &UID

Using &PI99 in Transfer Commands

If the remote Partner has sent its own value in Pi99 in the fpdu Ack-Create, this value is retained in the keyword &PI99. This keyword can be sent in a start of transfer or end of transfer command in the form &PI99.

Transfer commands are defined in the symbolic file definition associated with the transfer. For example, in the symbolic file definition FILE01, there is an end of transfer command defined as follows:

```
/S &PI99,D=&DS
```

If the remote Partner has sent the string string JOB002, the start command JOB002 is executed with the parameter DS=physical data set name received (without quotes) at the end of a successful transfer.

### Receiving 254 Characters in the Pi99 Field

The Generalized exit L1GFIUE1 enables you to receive 254 characters in the Pi99 field. Using the $PI99WD$ syntax shown below, you can create a sequential file with one 254-character record. The name of the data set is given as the parameter and it can contain dynamic variables.

```
$PI99WD$ PROD.&FILENAM.PI99.&REQNUMB
```

Receiving Requester

When the requester is the recipient, the monitor sends a reception request to a Partner for a remote file. The exchange of PeSIT fpdus during file selection is shown in the following diagram.
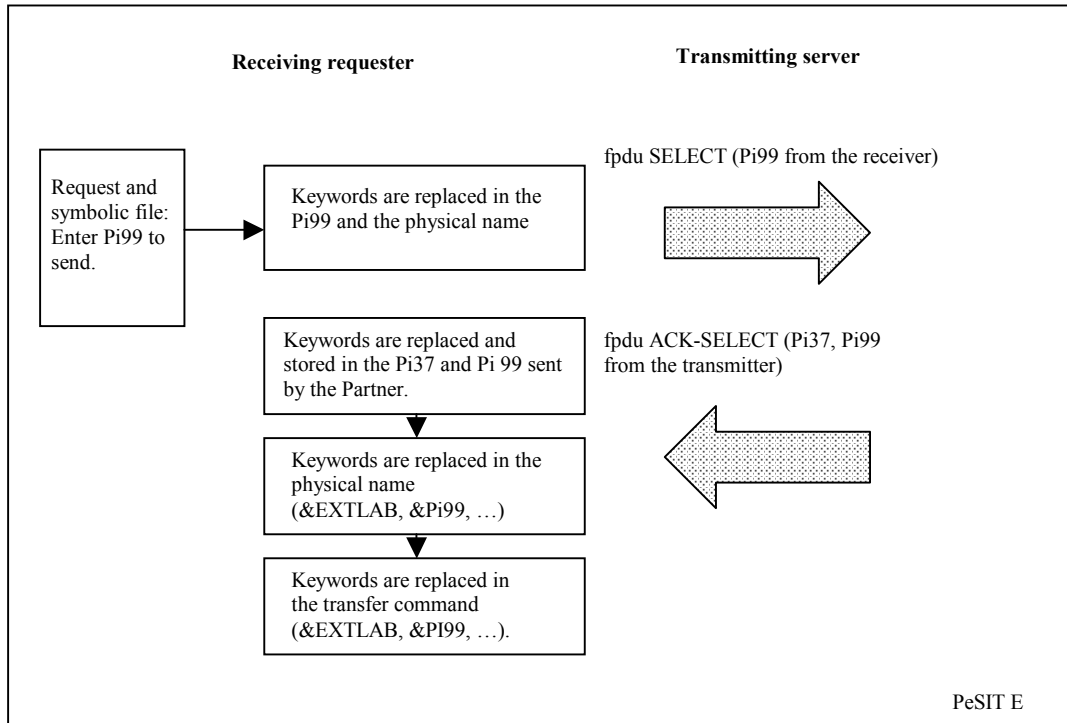
Note: With PeSIT D, the receiving requester does not send Pi99.

Transmitting Pi99 (PeSIT E)

When receiving, you can enter Pi99 in any of the following ways:

❖   In the symbolic file definition used by the transfer request. If you are using the user interface TSO/ISPF 1.2, enter this in the REMOTE DSN/PI99 field.

❖   In the 44-character REMOTE DSN/PI99 field when a request is entered with the user interface TSO/ISPF 4.1, Request Extension screen.

❖   In the parameter RDS= of the batch utilities P1B2PRQ2, P1B2PRQ3 and P1B2PREQ.

❖   In the 82-character API field, P: format when a request is entered with the user interface TSO/ISPF 4.1 Request Extension screen.

❖   In the parameter API=P: of the batch utilities P1B2PREQ and P1B2PRQ3.

❖   In the parameter API=P: of the batch utility P1B2PREQ.

❖   Using the $PI99$ keyword of the Generalized exit L1GFIUE1.

❖   Using the $PI99RD$ keyword of the Generalized exit L1GFIUE1.

Note: The REMOTE DSN and API fields cannot be used at the same time. Use The REMOTE DSN field if you need 44 characters, and the API field if you need 80 charcaters. Use the Generalized exit L1GFIUE1 (keywords $PI99$ and $PI99RD$) if you need 254 characters.

The value is placed in the Pi99 field that you want to send. If the value of Pi99 is not entered at the time of the transfer request, Connect:Express enters a default value from the symbolic file definition associated with the transfer. If no value is found, Pi99 is not sent. The Pi99 provided by the begining of transfer user exit replaces any string provided previously.

You can enter keywords in the Value field of Pi99. These keywords are replaced by the valid values when the fpdu Select is transmitted. See the Appendix for more information about keywords.

If the information exceeds the capacity of the Pi99 field, the string is truncated. The following variables can be used with Pi99:

&PARTNID, &FILENAM, &ORG, &DST, &REQNUMB, &REQDAT, &REQTIM, &DIR, &DSN, &UID

Receiving Pi37 and Pi99

You can use keywords in Pi37 and Pi99 sent by a Partner in the fpdu Ack-Select. These keywords are replaced by the valid values on the local system and the result is entered in the keywords &EXTLAB and &PI99, respectively. See the Appendix for more information about keywords.

If the information exceeds the capacity of the Pi99 field, the string is truncated. The following variables can be replaced in the Pi37 received from a Partner:

&PARTNID, &FILENAM, &ORG, &DST, &REQNUMB, &REQDAT, &REQTIM, &DIR, &DSN, &UID

The following variables can be replaced in the Pi99 received from a Partner:

&PARTNID, &FILENAM, &ORG, &DST, &REQNUMB, &REQDAT, &REQTIM, &EXTLAB, &DIR, &DSN, &UID

You can also send the keywords &EXTLAB and &Pi99 in the physical data set name and the transfer commands. You define transfer commands in the symbolic file definition associated with the transfer.
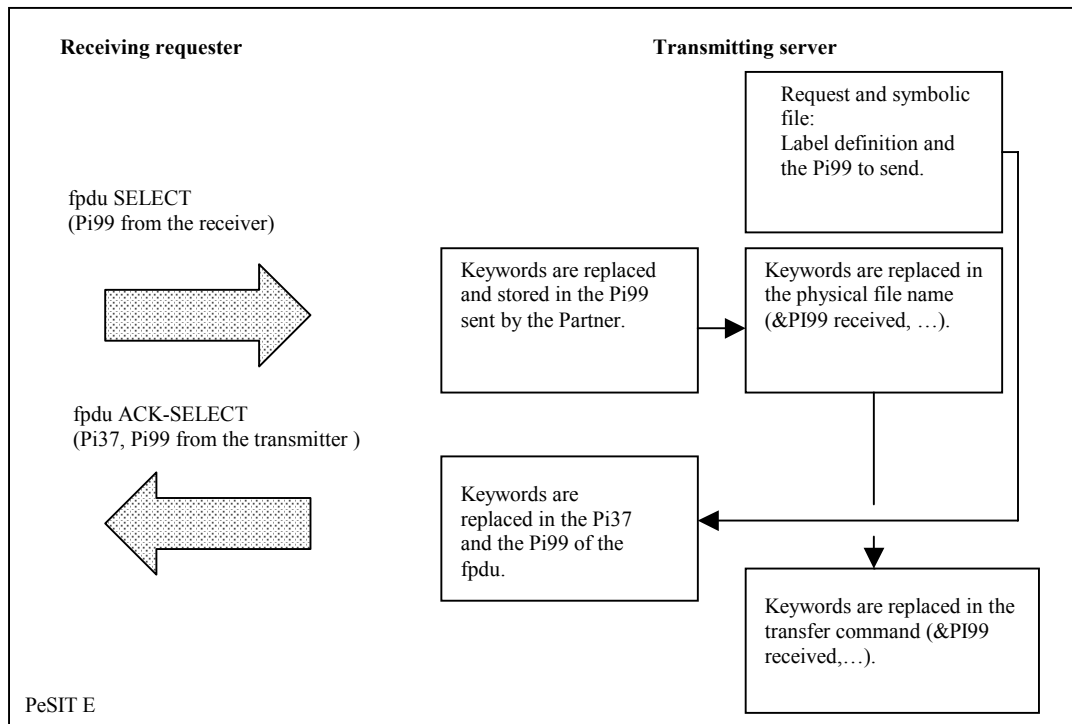
The Generalized exit L1GFIUE1 enables you to receive 254 characters in the Pi99 field. Using the $PI99WD$ syntax shown below, you can create a sequential file with one 254-character record. The name of the data set is given as the parameter and it can contain dynamic variables.

```
$PI99WD$ PROD.&FILENAM.PI99.&REQNUMB
```

Server

This chapter describes the use of Pi37 and Pi99 when a server transmits or receives file. The Pi99 sent by the requester contains complementary information for executing the transfer, and the Pi99 sent by the server contains complementary information associated with the server response.

Transmitting Server

When a server is transmitting a file, the monitor receives a reception request for a local file from a remote Partner. The following diagram shows how PeSIT fpdus are exchanged during the file selection phase:

Receiving requester	Transmitting server
	Request and symbolic file: Label definition and the Pi99 to send.
fpdu SELECT (Pi99 from the receiver)	Keywords are replaced and stored in the Pi99 sent by the Partner. / Keywords are replaced in the physical file name (&PI99 received, …).
fpdu ACK-SELECT (Pi37, Pi99 from the transmitter)	Keywords are replaced in the Pi37 and the Pi99 of the fpdu. / Keywords are replaced in the transfer command (&PI99 received,…).

PeSIT E

Note: With PeSIT D, the transmitting requester does not send Pi99.

Receiving Pi99 (PeSIT E)

The contents of Pi99 received from a remote Partner in the fpdu Select is stored in the keyword &PI99. If Pi99 contains keywords, they are replaced by the valid local values. You can use the following keywords when receiving Pi99:

&PARTNID, &FILENAM, &ORG, &DST, &REQNUMB, &REQDAT, &REQTIM, &DIR, &DSN, &UID

You can also use the keyword &PI99 in transfer commands. The Generalized exit L1GFIUE1 enables you to receive 254 characters in the Pi99 field. Using the $PI99WD$ syntax shown below, you can create a sequential file with one 254-character record. The name of the data set is given as the parameter and it can contain dynamic variables.

```
$PI99WD$ PROD.&FILENAM.PI99.&REQNUMB
```

Transmitting Pi37

You can enter Pi37 in any of the following ways:

❖   In the API field, U: format when a request type of hold is entered with the user interface TSO/ISPF 4.1, Request Extension screen

❖   In the parameter API=U: of the batch utilities P1B2PRQ2 and P1B2PRQ3

❖   In the parameters API=U: of the batch utility P1B2PREQ

❖   Using the $PI37$ keyword of the Generalized exit L1GFIUE1

By default, if the Label field is not entered with the request, then the physical file name is sent in Pi37. For implicit requests where the reception request is Normal and comes from an authorized remote Partner, the default label is always sent. If the information is more than 80 characters, the string is truncated.

You can also use keywords in the Label field. They are replaced by the valid values in Pi37 before the transmission of the fpdu Ack-Select. See the Appendix for more information about Keywords.

The following keywords can be used with Pi37:

&PARTNID, &FILENAM, &ORG, &DST, &REQNUMB, &REQDAT, &REQTIM, &DIR, &UID, &DSN

Transmitting Pi99

The Generalized exit L1GFIUE1 enables you to receive 254 characters in the Pi99 field. Using the $PI99WD$ syntax shown below, you can create a sequential file with one 254-character record. The name of the data set is given as the parameter and it can contain dynamic variables.

```
$PI99WD$ PROD.&FILENAM.PI99.&REQNUMB
```

When transmitting with a Hold request, you can enter Pi99 in any of the following ways:

❖   In the symbolic file definition used by the transfer request. If you are using the user interface TSO/ISPF 1.2, enter this in the REMOTE DSN/PI99 field.

❖   In the 44-character REMOTE DSN/PI99 field when a request is entered with the user interface TSO/ISPF 4.1, Request Extension screen.

❖   In the parameter RDS= of the batch utilities P1B2PRQ2, P1B2PRQ3 and P1B2PREQ.

❖   In the 82-character API field, P: format when a request is entered with the user interface TSO/ISPF 4.1 Request Extension screen.

❖   In the parameter API=P: of the batch utilities P1B2PREQ and P1B2PRQ3.

❖   In the parameters API=P: of the batch utility P1B2PREQ.

❖   Using the $PI99$ keyword of the Generalized exit L1GFIUE1.

❖   Using the $PI99RD$ keyword of the Generalized exit L1GFIUE1.

Note: The REMOTE DSN and API fields cannot be used at the same time. Use The REMOTE DSN field if
you need 44 characters, and the API field if you need 80 charcaters. Use the Generalized exit
L1GFIUE1 (keywords $PI99$ and $PI99RD$) if you need 254 characters.

If you do not use the Hold Request, you can only enter the description of Pi99 in the symbolic file definition
associated with the request. If you are using the user interface TSO/ISPF 1.2, enter the Pi99 description in the
REMOTE DSN/PI99 field. You can also use the $PI99$ and $PI99RD$ keywords of the Generalized exit
L1GFIUE1.

The value entered is placed in the Pi99 field that you want to send. If the value of Pi99 is not entered at the
transfer request level, Connect:Express looks for a value in the symbolic file definition associated with the
transfer. If no value is found, Pi99 is not transmitted. The Pi99 provided by the begining of transfer user exit
replaces any string provided previously.

You can also use keywords in the value field when entering data in Pi99. These keywords are then replaced by
the valid values in Pi99 before the transmission of the fpdu Ack-Select. See the Appendix for more information
about keywords.

If the information exceeds the capacity of Pi99, the string is truncated. You can use the following variables
with Pi99:

&PARTNID, &FILENAM, &ORG, &DST, &REQNUMB, &REQDAT, &REQTIM, &DIR, &DSN, &UID

Receiving Server

When the server is receiving a file, the monitor receives a transmission request for a remote file from a remote Partner. The following diagram shows the exchange of PeSIT fpdus during the file selection phase:
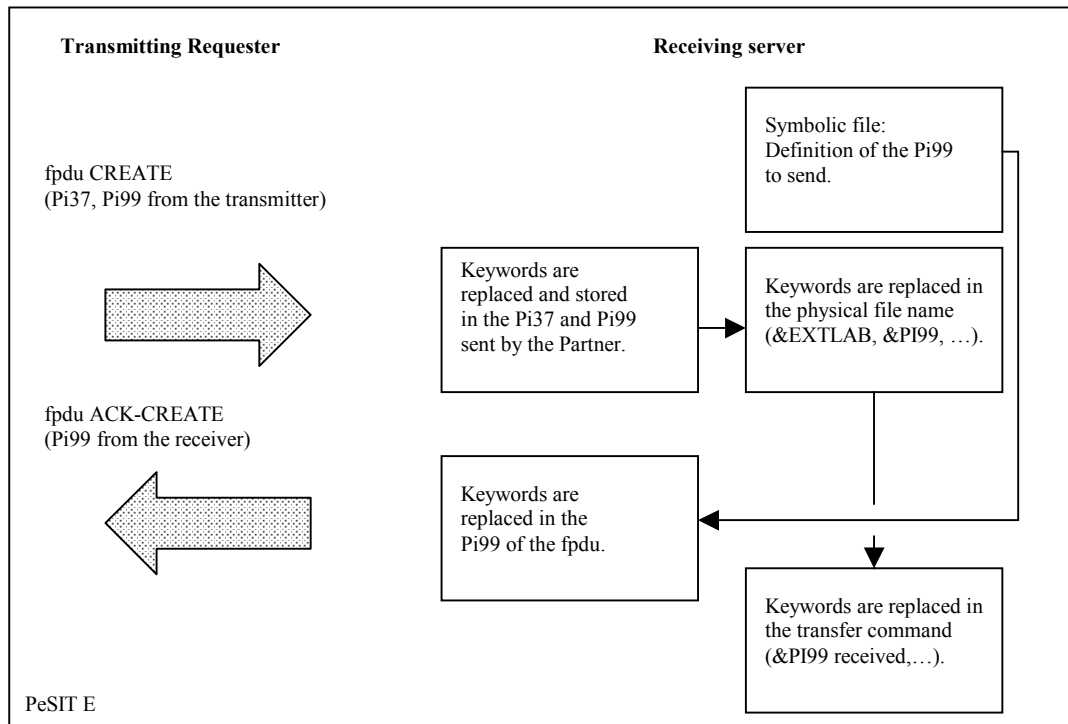
Note: With PeSIT D, the receiving server does not send Pi99.

Receiving Pi37 and Pi99

The contents of Pi37 and Pi99 received from a remote Partner in the fpdu Create are stored in the keywords &EXTLAB and &PI99, respectively. If Pi37 and Pi99 contain the valid syntax of a keyword, they are replaced by the valid local values. You can use the following keywords when receiving Pi37 and Pi99:

&PARTNID, &FILENAM, &ORG, &DST, &REQNUMB, &REQDAT, &REQTIM, &DIR, &DSN, &UID

The received Pi99 can also contain the keyword &EXTLAB. The keywords &EXTLAB and &PI99 can also be used in transfer commands.

The Generalized exit L1GFIUE1 enables you to receive 254 characters in the Pi99 field. Using the $PI99WD$ syntax shown below, you can create a sequential filewith one 254-character record. The name of the data set is given as the parameter and it can contain dynamic variables.

```
$PI99WD$ PROD.&FILENAM.PI99.&REQNUMB
```

Transmitting Pi99 (PeSIT E)

The receiver of the file can send a Pi99 to a Partner in the fpdu Ack-Create. This process is similar to transmitting Pi99 by the requester/transmitter, except that you only need to enter the description of Pi99 in the symbolic file definition associated with the request. If you are using the user interface TSO/ISPF 1.2, enter this

in the REMOTE DSN/PI99 field. You can also use the $PI99$ and $PI99RD$ keywords of the Generalized exit L1GFIUE1.

Appendix A

Keywords

This appendix lists the keywords that you can use in PeSIT user fields with a non-Connect:Express partner.

Keywords

The following table lists the keywords that you can use with a non-Connect:Express Partner.

Keyword	Maximum Length (characters)	Description
&PARTNID, &P	8	Symbolic Partner name
&FILENAM, &F	8	Symbolic File name
&ORG	8	Origin of the transfer request. Name of the partner originating the transfer request. (Pi3 of the PeSIT selection phase, truncated to 8 characters.)
&DST	8	Destination of the transfer request. Name of the destination Partner, or partner receiving the transfer. (Pi4 of the PeSIT selection phase, truncated to 8 characters.)
&REQNUMB	8	Request number in the format nnnnnnnn
&REQDAT	8	Date of transfer in the format Dyymmdd.
&REQTIM	6	Time of transfer in the format Hhhmmss
&EXTLAB	80	Contents of Pi37 (file label)
&DIR	1	Transfer direction. R - Reception T - Transmission
&UID	8	Name of the transfer requester
&DSN	44	Local physical file name
&PI99	44	Contents of Pi99

The following table lists the fields in which you can use keywords, and the possible keywords for each field.

Field	Location	Transfer Direction	Possible Keywords
Physical File Name	Transfer request	Transmission	&PARTNID, &FILENAM, &ORG, &DST, &DIR, &UID
		Reception	&PARTNID, &FILENAM, &ORG, &DST, &REQNUMB, &REQDAT, &REQTIM, &DIR, &DSN, &UID, &EXTLAB, &PI99
Physical File Name	Symbolic File definition	Transmission	&PARTNID, &FILENAM, &ORG, &DST, &DIR, &UID
		Reception	&PARTNID, &FILENAM, &ORG, &DST, &REQNUMB, &REQDAT, &REQTIM, &DIR, &DSN, &UID, &EXTLAB, &PI99
Pi99	Transfer request	Transmission	&PARTNID, &FILENAM, &ORG, &DST, &REQNUMB, &REQDAT, &REQTIM, &DIR, &DSN, &UID
		Reception	&PARTNID, &FILENAM, &ORG, &DST, &REQNUMB, &REQDAT, &REQTIM, &DIR, &DSN, &UID, &EXTLAB
Pi99	Symbolic File definition	Transmission (Requester)	&PARTNID, &FILENAM, &ORG, &DST, &REQNUMB, &REQDAT, &REQTIM, &DIR, &DSN, &UID
		Reception (Server)	&PARTNID, &FILENAM, &ORG, &DST, &REQNUMB, &REQDAT, &REQTIM, &DIR, &DSN, &UID, &EXTLAB
		Reception (Requester)	&PARTNID, &FILENAM, &ORG, &DST, &REQNUMB, &REQDAT, &REQTIM, &DIR, &DSN, &UID
		Transmission (Server)	&PARTNID, &FILENAM, &ORG, &DST, &REQNUMB, &REQDAT, &REQTIM, &DIR, &DSN, &UID
&PI99	Received fpdu Create	Reception	&PARTNID, &FILENAM, &ORG, &DST, &REQNUMB, &REQDAT, &REQTIM, &DIR, &UID, &EXTLAB
&PI99	Received fpdu Ack-Select	Reception	&PARTNID, &FILENAM, &ORG, &DST, &REQNUMB, &REQDAT, &REQTIM, &DIR, &UID, &EXTLAB
API=U	Transfer request	Transmission	&PARTNID, &FILENAM, &ORG, &DST, &REQNUMB, &REQDAT, &REQTIM, &DIR, &DSN, &UID
&EXTLAB	Received Pi37 of the fpdu Create	Reception	&PARTNID, &FILENAM, &ORG, &DST, &REQNUMB, &REQDAT, &REQTIM, &DIR, &UID
&EXTLAB	Received Pi37 of the fpdu Ack-Select	Reception	&PARTNID, &FILENAM, &ORG, &DST, &REQNUMB, &REQDAT, &REQTIM, &DIR, &UID
Start of transfer Command	Symbolic File definition	Transmission	&PARTNID, &FILENAM, &ORG, &DST, &REQNUMB, &REQDAT, &REQTIM, &DIR, &UID, &PI99
		Reception	&PARTNID, &FILENAM, &ORG, &DST, &REQNUMB, &REQDAT, &REQTIM, &DIR, &UID, &EXTLAB, &PI99

Field	Location	Transfer Direction	Possible Keywords
End of transfer Command	Symbolic File definition	Transmission	&PARTNID, &FILENAM, &ORG, &DST, &REQNUMB, &REQDAT, &REQTIM, &DIR, &UID, &PI99
		Reception	&PARTNID, &FILENAM, &ORG, &DST, &REQNUMB, &REQDAT, &REQTIM, &DIR, &UID, &EXTLAB, &PI99

Keyword Syntax

Keywords can take any of the following forms. In the following examples, VAR represents the keyword name.

Syntax	Description
&VAR	The keyword is replaced by the value represented by VAR.
&position:lengthVAR	The keyword is replaced by the substring extracted from the value represented by VAR, based on the position and length that you specify. For example, if you specify position 0, this indicates the first byte.
&lengthVAR	The keyword is replaced by the substring extracted from the value represented by VAR, based on the first byte and the length that you specify.
&position:VAR	The keyword is replaced by the substring extracted from the value represented by VAR, based on the position that you specify.

For example, **&PI99** is replaced with the contents of Pi99, and **&5:16PI99** is replaced by the text string extracted from Pi99 in the 5th byte with a length of 16.

If the replaced value is less than the position and length that you specify, spaces are added in the final positions. If the position is greater than the length of the value represented by VAR, &VAR is replaced with an empty string. In the following examples, VAR = 20-character string.

Text string	Result
beg&*VAR*end	beg*20-character string*end
beg&*13:10VAR* end	beg*string* end (four spaces were added)
beg&*30VAR* end	begend

If VAR does not correspond to a known keyword, there is no substitution and &VAR results in the text string "&VAR". For example, AA&UNKNOWNbb remains AA&UNKNOWNbb. However, if VAR is a known variable and you want to insert the string "&VAR" without substituting the value for the keyword, then you double the &. For example, &&VAR results in the string &VAR.

Text string	Results
&&PI99	&PI99
&ABC	&ABC

Text string	Results
&&ABC	&&ABC
&:VAR &0:VAR &0:0VAR &:0VAR &0VAR	&VAR
&0:xVAR	&xVAR

In the following examples, VAR represents the text string ABCDEFGH.

Text string	Is replaced with
&VAR	ABCDEFGH
XY&VARZ	XYABCDEFGHZ
&3VAR	ABC
&0:3VAR	ABC (Same as &3VAR)
&10VAR	'ABCDEFGH ' (2 spaces at the end)
&<10VAR	'ABCDEFGH' (No spaces at the end)
&4:10VAR	'EFGH ' (6 spaces at the end)
&4<10VAR	'EFGH' (No spaces at the end)
&3:4VAR	DEFF
&5:VAR	EFGH
&10:VAR	" "(empty string)
&0:0VAR	ABCDEFGH (Same as &VAR)

Appendix B

Using Environment Variables

This appendix describes how you can define long names for files and IP hosts.

Why to Use Long Names

Supporting HFS files means that file names can be much more than 44 characters long. This problem applies to *IP HOST* names, limited to 32 characters long as a Connect:Express parameter fields, and to the *Remote dsn / Pi99* , limited to 44 characters long as a Connect:Express parameter fields.

Connect:Express uses environment variables to support long names. The environment variables are specfic to the monitor, and they are resolved when the name is processed. You can define variables in a file allocated to TOM address space using the ENVVAR DD card . All address spaces that need to process environment variables , such as the APM, the ANM, the AFM and the EAS, know this file.

The fields *Dsn local*, *IP HOST* and *Remote dsn/Pi99* can be extended up to 256 characters. They can contain several variables.

ENVVAR Environment Variables File

The environment variables file is defined in TOM JCL, using DD ENVVAR card. This is a sequential file, fixed format, record length less equal 1024 characters.

```
//SYSIN    DD      DISP=SHR,DSN=PROD.CEXPRESS.SYSIN(TOM4)
//SYJNL    DD      DISP=SHR,DSN=PROD.CEXPRESS.TOM4JNL
//SYLOG    DD      SYSOUT=V,HOLD=YES
//ENVVAR   DD      DISP=SHR,DSN=PROD.CEXPRESS.ENVVAR(TOM4)
```

The environment variable process can issue WTO messages prefixed by 'ZVAR', as shown in the example below, where the error is that the file record format is variable.

```
ZVAR004E OPEN ENVVAR ERROR  RECFM: 00000080
```

These messages can be issued by any address space that processes variables, APM , ANM, AFM ou EAS. When a syntax error is detected, the process issues a warning and continues. The following example shows that a syntax error has been detected at line 00003 of the ENVVAR file: no '>' character has been found.

```
ZVAR006W ENVVAR SYNTAX ERROR   :   NO >    00003
```

If the variable cannot be resolved due to a syntax error, or because no corresponding definition exists, the process continues without resolving it.

Option 0 of TSO/ISPF interface enables the operator to access the ENVVAR file. Use option'S' to edit the file, updates are dynamically available.

```
 TOM4230-------------- INITIALIZATION 2/2 ------------------------------------
  OPTION ===> ?                                          2009/04/01 07:27:27
  4XX/TEST
             ?  MONITOR ===> TOM3  NAMES INITIALIZATION ( AUTOMATIC --> YES ).
              TEMPORARY WORK-UNIT ===> SYSDA    , JES2-INTERFACE ---> ISF
             L LOGON-PROCEDURE, O  OPTIONS, V ISPF INSTALLATION CHECKING.
 ---------  S : CHECK FILES OF JOBTOM3   CSGB ACTIVE GLOBAL
 V
 _  ISPLLIB    ===> PROD.CEXPRESS.ISPLLIB
 _  LOADLIB    ---> PROD.CEXPRESS.LOADLIB.PSR0003
 _             --->
 _  SYSSNA       -> PROD.CEXPRESS.SYSPRM(L4SNA)
 _  SYSX25       -> PROD.CEXPRESS.SYSPRM(L4X25)
 _  SYSTCP       -> PROD.CEXPRESS.SYSPRM(L4TCP)
 _  SYSUE1       -> PROD.CEXPRESS.PARMLIB(SYSUE1)
 _  SYSCE1       ->

 S  ENVVAR       -> PROD.CEXPRESS.ENVVAR(TOM30)
 _  ANMSSL       ->
 _  AFMFTPE      ->

              X EXIT, -PF3- FIN, -PF10/11- DEFILEMENT
```

Syntax and Processing of the Variables

To define a variable in ENVVAR file, use the following syntax :

```
<var>value to replace
```

The string 'var' can contain 1 to 8 characters , including blanks, excluding '>'. The string <var> must be placed in the name you want to extend : 'value to replace' will replace it when processed.

Variables are processed in the following situations :

1. •Transfer of an HFS file – data set name processed in APM for PeSIT and Odette, or EAS for Ftp.

2. •TCP/IP connection with host name – processed in ANM or AFM depending on the protocol.

3.   •Sending and receiving PeSIT PI99 – processed in APM.

4.   •Showing the variable in the TSO/ISPF screens (File definition, Partner definition, Journal)

Message	Description
ZVAR004E OPEN ENVVAR ERROR	The ENVVAR record format is invalid. The cause is indicated. The file transfer is not interrupted, but variables are not resolved
ZVAR006W ENVVAR SYNTAX ERROR	While processing, a syntax error has been detected – the error type and the line number are indicated. All possible variable resolution are completed.
ZVAR007W OVERFLOW, STRING TRUNCATED	While processing, an overflow has been detected : the result is more than 256 characters long : the result is truncated, the process stops.
ZVAR009W VARIABLE NOT FOUND	The variable shown in the message has no definition in the ENVVAR file. All possible resolution are completed

Examples of Use

You can use environment variables in the Dsn local field of the symbolic file definition for an HFS file ,in the IP HOST field of the partner definition for a TCP/IP partner , and in the Remote Dsn/Pi99 field of the file definition or the transfer request . You can request the variable resolution in the screen using a command VAR.