

SYSTEM 2000[®] Software: Interface to CICS[®]

Version 1
First Edition

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SYSTEM 2000® Software: Interface to CICS®, Version 1, First Edition*, Cary, NC: SAS Institute Inc., 2000.

SYSTEM 2000® Software: Interface to CICS®, Version 1, First Edition

Copyright © 2000 by SAS Institute Inc., Cary, NC, USA.

ISBN 1-58025-605-8

All rights reserved. Produced in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

U.S. Government Restricted Rights Notice. Use, duplication, or disclosure of the software and related documentation by the U.S. government is subject to the Agreement with SAS Institute and the restrictions set forth in FAR 52.227-19 Commercial Computer Software-Restricted Rights (June 1987).

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, August 2000

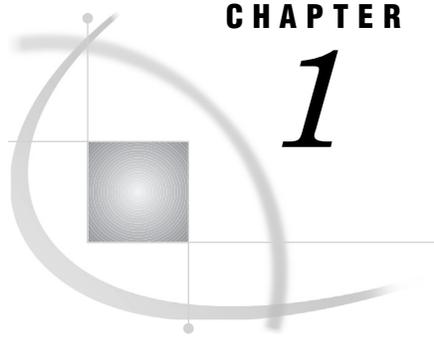
SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries.® indicates USA registration.

IBM® and all other International Business Machines Corporation product or service names are registered trademarks or trademarks of International Business Machines Corporation in the USA and other countries.

Other brand and product names are registered trademarks or trademarks of their respective companies.

Contents

Chapter 1	△ Introduction	1
Overview		1
Compatibility		2
S2KCUSE		2
Chapter 2	△ Supplied Transactions	3
Introduction		3
Chapter 3	△ PLEX Processing	15
Introduction		15
Precompiler Options		15
STOP S2K Processing		16
Pseudo-Conversational Transactions		16
COBOL Differences		17
Chapter 4	△ Error Recovery and Security	19
Introduction		19
Error Recovery		19
Shutdown Logic		20
Chapter 5	△ User Exits	27
Introduction		27
CICS User Exits		27
Non-CICS User Exits		30
Index		31



CHAPTER

1

Introduction

<i>Overview</i>	1
<i>Compatibility</i>	2
<i>S2KCUSE</i>	2

Overview

The SYSTEM 2000 interface to CICS allows direct communication between the CICS transaction processing system and SYSTEM 2000 Multi-User data management software. This document discusses how to use the interface to CICS for Self-Contained Facility (SCF) sessions, Programming Language Extension (PLEX) jobs, and error recovery, and provides reference information about supplied SCF transactions.

The interface to CICS enables you to access SYSTEM 2000 databases by using product-supplied SCF transactions or user-written PLEX CICS transactions.

The supplied transactions include an SCF command editor that enables 3270-type terminals to create, edit, and save SCF command streams. Also, several user exits are provided in the interface. These exits can be used to journal SCF input, to adjust accounting ID fields, and to perform special message editing. See Chapter 5, “User Exits,” on page 27, for more information.

By using the SYSTEM 2000 PLEX language, you can write your own SYSTEM 2000 CICS transactions. User communication with the interface in PLEX differs from SCF because communication with the terminal is entirely up to the PLEX program. The interface provides the communication protocol and controls. See Chapter 3, “PLEX Processing,” on page 15, for more information.

CICS users share the databases with batch and other TP users. Each site may have multiple CICSs and Multi-User software running concurrently, however, each CICS may communicate with only one Multi-User region.

The four functions of the SYSTEM 2000 interface to CICS are

- SCF processing
- PLEX processing
- operator communication processing
- error recovery processing.

Compatibility

SYSTEM 2000 software is fully compatible with CICS Releases 4.1 through CICS TS. However, Version 1 of SYSTEM 2000 supports only the command-level interface, and Release 12.1 is the last release of this software to support the macro-level interface.

Note that programs that are not COBOL II programs must have the PLXPBLD program link edited with the PLEX program, and PLXPBLD must be from the same release as the precompiler, or unpredictable results can occur.

S2KCUSE

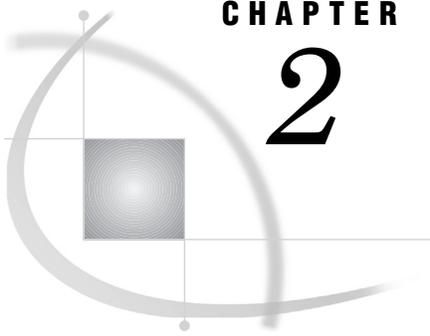
The CICS resident module S2KCUSE contains the options that define or govern the interface in the CICS environment. S2KCUSE can be modified to fit the needs at your site.

Two options that CICS PLEX programmers must be aware of are DOSTOP= and CHKTRNS=. Both of these options work specifically with automatic STOP S2K processing. They should be examined closely because incorrect settings can adversely affect system performance. STOP S2K processing is described in greater detail in Chapter 3, "PLEX Processing," on page 15.

The option DOSTOP= controls whether automatic STOP S2K processing is active. The default is Y (Yes). Sites that do their own STOP S2K commands or that always use START S2K USE(2) should change DOSTOP=Y to DOSTOP=N (No).

The option CHKTRNS= adds an additional level of checking for pseudo-conversational transactions. The default is CHKTRNS=N, which means that the interface executes a STOP S2K at task termination. If you change CHKTRNS=N to CHKTRNS=Y, an additional check is performed to see if your terminal is locked to another transaction. If the next transaction ID is set, then the STOP S2K is not issued. If you do the EXEC CICS RETURN without the TRANSID, then the interface will execute a STOP S2K for you.

See the *SYSTEM 2000 Installation Instructions*, which you received with the Version 1 software, for details about S2KCUSE.



CHAPTER

2

Supplied Transactions

<i>Introduction</i>	3
<i>S2KU</i>	4
<i>S2KK</i>	5
<i>S2KE</i>	7
<i>S2OP</i>	12
<i>LALL, LPLX, and LSCF</i>	12
<i>TALL, TPLX, and TSCF</i>	14
<i>UDMP</i>	14

Introduction

The SYSTEM 2000 interface to CICS enables you to access SYSTEM 2000 databases by means of supplied SCF transactions or user-written PLEX CICS transactions. This chapter describes the transactions supplied in SYSTEM 2000. See Chapter 3 for the information that you need in order to write your own transactions.

The supplied transactions and their functions are

S2KU	activates the interface.
S2KK	assigns program function keys.
S2KE	invokes the SCF Command Editor.
S2OP	invokes the operator communication transaction.
LALL, LPLX, and LSCF	list the status of users in the interface.
TALL, TPLX, and TSCF	terminate users in the interface.
UDMP	produces a CICS transaction dump of S2KCUSE.

S2KU

Activates the **SYSTEM 2000 CICS** interface.

Syntax

S2KU

S2KU USER, *password*:

Details

To activate the interface at a CICS terminal, type S2KU. The S2KU transaction displays a sign-on screen that is similar to Display 2.1.

Display 2.1 S2KU Sign-on Screen

```

                SYSTEM 2000 CICS INTERFACE
                SELF-CONTAINED FACILITY

                PLEASE ENTER YOUR SIGN-ON INFORMATION

                USER PASSWORD ..... :
                DATABASE NAME TO ACCESS OR EXIT ..... : employee_____
                ARE YOU CREATING A NEW DATABASE ?..... : NO_ (NO/YES OPTIONAL)
                DO YOU NEED EXCLUSIVE USE OF THE DATABASE? : NO_ (NO/YES OPTIONAL)

                WHEN ALL INFORMATION IS ENTERED PRESS ENTER KEY
  
```

In the sign-on screen, **USER PASSWORD** and **DATABASE NAME** are required fields unless you are exiting from this screen. To exit, type **EXIT** in the **DATABASE NAME** field and press **ENTER**.

You can bypass the sign-on screen by typing **S2KU** followed by the **SCF USER** command and then typing your database password. This locks your terminal to the S2KU transaction until you issue the **EXIT** command.

A CICS SCF session contains a series of SCF commands. S2KU processes input and output, one screen at a time. It accepts one screen transmission of SCF commands. That is, an SCF command cannot span screens, although the SCF Command Editor (S2KE) can be used to build and store SCF command streams as members in the Command Editor libraries. You can recall the stored members for editing and submit them to Multi-User for execution. An output page manager receives the output from Multi-User, formats the data into screen pages, and stores the pages using CICS temporary storage.

You can enter the paging commands that follow to browse the stored pages.

P,E	enable paging
P,D	disable paging
P,T	display page statistics
P,A	display all remaining pages
P,N	display next page
P,P	display previous page
P,C	display current page
P,F	display first page
P,L	display last page
P, <i>n</i>	display page <i>n</i>
P,+ <i>n</i>	display page (current page + <i>n</i>)
P,- <i>n</i>	display page (current page - <i>n</i>)
P,LMAX= <i>nnnn</i>	sets the number of output lines allowed. The maximum is 2000.
P,PAGE= <i>nnn</i>	sets the number of lines per page (non-3270 terminals only). The maximum is 255.
P,LINE= <i>nnn</i>	sets the number of characters per line (non-3270 terminals only). The maximum is 255.

If you press the ENTER key without entering a paging command, the next page appears. The SCF session is non-conversational. It returns control to CICS after displaying a response.

S2KK

Assigns program function keys.

Syntax

S2KK <*environment*>

Details

Default PF key values are set at installation time. However, by using the S2KK transaction, PF key values can be set for individual users. If the user can be associated with an operator ID, the operator ID becomes the “key” for preserving the PF key changes. If no operator ID is available, the terminal ID is used as the “key” for the PF key changes.

The PF key changes that you indicate by using the S2KK transaction are maintained in the Command Editor data set S2KLIB. If no S2KLIB is defined, temporary storage is used. While the temporary storage file remains intact, the PF key settings are good. If the file is initialized by CICS, the PF key settings are lost. PF key settings are preserved in the S2KLIB data set until the operator who assigned them either changes them again or deletes them.

The S2KK transaction can be executed during an S2KU session, an S2KE session, an S2OP session, or as a stand-alone transaction. When S2KK is used during one of these sessions, no parameters are accepted, and it is assumed the changes are for the PF keys of that session. For example, if issued during an S2KU session, the S2KK transaction can change only the SCF PF key settings. If issued during an S2KE session, the S2KK transaction can change only the PF key settings for the Command Editor. When issued as a stand-alone transaction, S2KK can change or delete the PF key settings for any of the three environments: S2KU, S2KE, S2OP.

The DELETE option is used during a session to revert to the installed defaults for that environment. The previously preserved PF key settings are deleted from either the S2KLIB data set or from temporary storage so that the default settings are used.

The result of issuing the S2KK transaction, as just described, is a formatted display screen that contains PF keys 1 through 24, PA keys 1 through 3, and the CLEAR key. All keys that are assigned default values have those values displayed. All keys that do not have a default value are blank. Each field allows up to 32 characters of input as the PF key setting. Settings for the S2KU PF keys can be

- paging commands
- valid SCF syntax
- the S2KE verb.

Settings for the Command Editor can be valid S2KE syntax only. When the S2KE verb is used as an S2KU PF setting, the interface executes the PF key values from both environments. That is, if S2KU PF10 has the value **S2KE** and S2KE PF10 has the value **INSERT**, pressing PF10 in S2KU opens the Command Editor and puts it in insert mode. If S2KU PF11 and PF23 have the value **S2KE** and S2KE PF11 and PF23 have the value **MODIFY**, then pressing either PF11 or PF23 in S2KU opens the Command Editor in modify mode. The settings in the previous examples are defaults. You can use S2KK to define additional functions, such as executing the last command in your work buffer or executing the same S2KE member multiple times during the session (set an S2KU PF key to **S2KE** and its S2KE counterpart to **SEND** or **SEND MEMBER=XXXXXX**, respectively). Incompatible PF key settings between the environments can result in conflicts.

Settings for the S2OP environment are limited to valid paging commands. The PF keys 3 and 15 are permanently assigned as end-of-session commands in the S2OP environment. Values assigned to these two PF keys are ignored in the S2OP environment.

In some instances, the PA keys have been reserved by system facilities and are not passed on to the executing program. For example, the use of PA keys to switch to multiple sessions on a terminal. In this case, the PA key is not passed to the program, therefore, any value assigned to this key is never processed by the program.

Command Editor users should be aware that when the S2KK transaction is issued by many users, it can impact the amount of file space that is available for Command Editor members.

The S2KK transaction is a conversational transaction. It does not release control when waiting for a response from the terminal. Any attention identifier (AID) other than the ENTER key causes the transaction to terminate. If S2KK is issued from the S2KU, S2KE, or S2OP environments, control is returned to that session. If issued from a stand-alone environment, a message that indicates end-of-processing is issued, and the terminal is released from the S2KK transaction. If PF keys are set for a terminal (no operator sign-on), anyone who uses that terminal, also without an operator sign-on, will be using those PF key settings unless they change them by using S2KK.

S2KE

Invokes the SCF Command Editor.

Syntax

S2KE

S2KE LIBRARY=*library-name* MEMBER=*member-name* <READPW=*read-password*>
<WRITEPW=*write-password*>

Details

The S2KE transaction invokes the Command Editor. You can initialize the Command Editor as either an independent session or a subsession within a regular SCF session (S2KU). Both methods use the same syntax.

You may specify both a library and a member in that library to be edited. If you do not specify a library name, the default is S2KLIB. A work file is used if a member is not specified. If you specify a member name, that member is copied to the work file. All editing is performed on the contents of the work file. When you issue a SAVE command, the work file replaces the original library member. During Command Editor operations, the specified member is not locked or flagged when it is being updated.

Command Editor commands are grouped into three categories:

- control commands
- library and member commands
- text editing commands.

The commands available in each group are shown in Table 2.1.

Table 2.1 Commands Provided by S2KE

Control Commands	Library and Member Commands	Text Editing Commands
END	GET	BOTTOM
SEND	LISTD	DELETE
SUBMIT	SAVE	DOWN
S2KE	SCRATCH	FIND
S2KK		INSERT
		MODIFY
		TOP
		UP

An S2KE command can use a maximum of 80 columns. Command Editor syntax requires a blank between a command verb and a parameter. You can use a comma or one or more blanks to separate parameter specifications.

Here are additional details about the Command Editor.

Command Editor Overview The Command Editor is a line-oriented editor that operates on the work file concept. You are assigned a private work file that you can modify without interference from other users. In the work file, the Command Editor maintains a line pointer to indicate the current line that is being edited. Changes made during a Command Editor session are written to the work file only. When you issue the SAVE command, the Command Editor writes the changes to a Command Editor library. The available libraries are defined at installation time, and they include a library that has the default name S2KLIB.

Command Editor Libraries and Members A Command Editor library is a key-sequenced VSAM file that is defined to CICS. A unit of data in a library and owned by a user is called a member of the library. Users share a Command Editor library, and protection for each member is provided by optional READ and WRITE passwords.

A Command Editor member is a group of 80-character lines of text padded with trailing blanks, which is entered from a terminal. Access to a member is based on its 8-character member name, which must be unique within a library.

User Identification When you initialize a Command Editor session, various resources are assigned to you. Some resources are retained only during that session, but others are extended to the next session. If appropriate user identification is supplied, such as operator ID, the Command Editor is able to uniquely identify users. As a result, resources can be maintained from session-to-session.

SYSTEM 2000 attempts to locate a unique user identifier by looking for an ID in this order:

- 1 Operator ID - supplied by the sign-on transaction or by pre-set terminal security
- 2 Terminal ID.

If SYSTEM 2000 is forced to use the terminal ID as an identifier and you change terminals between Command Editor sessions, you will receive what was left in the work file by the last user of that terminal.

Control Commands Control commands enable you to initialize or terminate a Command Editor session and to define or display values for function keys. Here are the control commands listed in alphabetical order.

END

terminates the current Command Editor session and returns control to the previous environment. If S2KE was entered from an S2KU session, the END command causes a return to S2KU and a display of the current page (P,C).

SEND <LIBRARY=*library-name*> <MEMBER=*member-name*>
<READPW=*read-password*>

terminates the Command Editor session, submits the contents of the specified member or the work file to Multi-User through the interface, and passes control to an SCF session. The submitted command stream is logged onto the work file and replaces its current contents.

SUBMIT <LIBRARY=*library-name*> <MEMBER=*member-name*>

submits JCL to the OS/390 (MVS) system for batch execution. The JCL can be stored in the work file or in a member. It is sent to the destination ID that was configured during installation.

S2KE <LIBRARY=*library-name*> <MEMBER=*member-name*>
<READPW=*read-password*> <WRITEPW=*write-password*>

initializes a Command Editor session as an independent session or as an SCF (S2KU) sub-session.

If you omit the MEMBER parameter, the Command Editor opens the primary Command Editor screen and operates on the current contents of the work file. When issued with the MEMBER parameter, the S2KE command reads the specified member into the work file or creates the member if one does not already exist. If a member is password protected, the appropriate password is required.

S2KK

See the section “S2KK” on page 5.

Library and Member Commands The library and member commands enable you to create, display, remove, and update members in a library. Here are the commands listed in alphabetical order.

GET <LIBRARY=*library-name*> <MEMBER=*member-name*>
<READPW=*read-password*> <TYPE=*member-type*>

inserts the contents of the specified member into the work file following the current position. You must specify a READ password if the member is read-protected. The last inserted line becomes the current line.

LISTD <LIBRARY=*library-name*> <START=*member-name*>
<MEMBER=*member-name*> <TYPE=*member-type*>
<READPW=*read-password*>

displays the directory of the specified library in alphabetical order. LISTD with no operands displays the directory of the current library. The other operands limit the display as follows:

START	begins the directory listing that contains the specified member.
MEMBER	displays only the directory entry of the specified member.
TYPE	limits the display to members of the indicated type. In the case of TYPE=PFKY, only the fields that contain member name and type are filled in on the display.

LISTD does not display members that have 8-character names ending in TEMP and having a type of S2K. These are considered default members and are not for general display. If the member is read-protected, a password is required.

```
SAVE <LIBRARY=library-name> <MEMBER=member-name>
      <READPW=read-password> <TYPE=member-type>
      <WRITEPW=write-password>
```

replaces the specified member with the contents of the work file. You must specify a WRITE password if the member is write-protected. If the member does not exist in the specified library, a new member is created. You can also specify READPW, TYPE, and WRITEPW protections.

If you omit the MEMBER parameter, the current member is updated. If you omit READPW or WRITEPW when you save a new member, the password is set to blanks.

```
SCRATCH <LIBRARY=library-name> <MEMBER=member-name>
        <TYPE=member-type> <WRITEPW=write-password>
```

removes the specified member from the library. You must specify a WRITE password if the member is write-protected. You can also specify TYPE protection.

If you omit the MEMBER parameter, the current member is removed.

Text Editing Commands You can edit lines of text by issuing commands that indicate the current line of text. Then, you issue the commands that delete, modify, or insert lines of text. A line is recognized by its relative position in the work file. The Command Editor maintains a line pointer to indicate the current (target) line.

The display screen for the Command Editor has three areas, as shown in Display 2.2.

Display 2.2 S2KE Display Screen

```
EDIT      MEMBER=0019TEMP TYPE=S2K  LIBRARY=S2KLIB  TOTLINES=0001  CURLINE=0001
1234567890...15...20...25...30...35...40...45...50...55...60...65...70...75...80
EXIT:
*****END OF FILE*****
```

- The *Input Area* is the top line on the screen; it holds a maximum of 80 characters. This area contains user-specified commands and messages to the Command Editor. If the input is shorter than 80 characters, the Input Area is padded with trailing blanks.
- The *Status Area* is the third line on the screen; it displays status information.
- The *Text Display Area* is the remainder of the screen; it shows the contents of the work file. The current line appears at the top of this area.

The text editing commands that follow enable you to position the current line to the top or to the bottom of the work file, up or down (relative to the current line), or to the line that contains a specific string.

BOTTOM

positions the line pointer at the bottom of the file.

DELETE <*n* | *>

deletes *n* lines from the file, beginning with the current line. The line pointer is then positioned at the next remaining line. An asterisk (*) deletes all lines from the current position to the end of the file. The default is 1 line.

DOWN <*n*>

moves the line pointer down *n* lines. The default is 1 line.

FIND *string*

finds the next occurrence of the specified string and positions the line pointer at the line that contains the string. You can use quotes, slashes, or blanks as delimiters for the string, however, the delimiter must be a character that does not appear within the string.

INSERT *n*

inserts *n* lines into the work file below the current position. The inserted lines are taken from the text in the Input Area, one at a time. If you omit *n*, insert mode continues until the Command Editor encounters a null line. The last inserted line becomes the current line. If you do not specify *n*, you can insert any number of lines. You must submit a null line before returning to edit mode.

MODIFY

displays the current line in the Input Area for modification. You can use the line-pointer commands and the DELETE and INSERT keys on the 3270 terminal. To make the modification,

- 1 Display the line you want to modify on the first line of the Text Display Area.
- 2 Enter the MODIFY command, which copies the line into the Input Area.
- 3 Use the terminal editing keys to modify the line displayed in the Input area.
- 4 Press ENTER, which replaces the original line in the work file with the modified text from the Input Area. The modified line cannot be more than 80 characters in length.

TOP

positions the line pointer at the top of the file.

UP <*n*>

moves the line pointer up *n* lines. The default is 1 line.

Command Logging The Command Editor logs the last SCF command in the work file. If the system fails and the next user of this terminal enters S2KE as the first transaction, the last command issued by the previous user is seen. If the system fails after you issue the SCF commands USER and DATA BASE NAME IS but before you enter any other information, then the password and database name may be revealed.

S2OP

Invokes the CICS operator communication transaction.

Syntax

S2OP *multiuser-console-command*

Details

The S2OP transaction is a stand-alone transaction. It cannot be executed from an S2KU session. Any SYSTEM 2000 Multi-User operator command that can be entered from the computer console can be issued by using the S2OP transaction. If the number of lines of response add up to more than one page of output, the terminal is locked to the S2OP transaction.

Console operator commands that are sent through the interface by using the S2OP transaction are processed like SCF commands. Output is returned to the terminal in 4K blocks. For any command, you can receive up to 8000 bytes of messages that reflect the status of the system when the command was processed. If there are more than two blocks of output, the remainder might reflect changes that occurred in the system while the first two output blocks were being processed by the interface. A line of asterisks identifies the second block.

The terminal that issues a console operator command appears as an active user in the output for a DISPLAY ACTIVE console operator command. The user type is ALT CON.

S2OP has its own set of PF keys that can be modified by using S2KK. The PF3 key and the PF15 key terminate the S2OP session.

LALL, LPLX, and LSCF

List the status of PLEX and SCF users in the interface.

Syntax

LALL <USERID=*userid*>

LPLX <USERID=*userid*>

LSCF <USERID=*userid*>

Details

The LALL transaction lists the status of all SYSTEM 2000 users currently occupying resources in the interface. A sample LALL transaction is shown in Output 2.1.

Output 2.1 LALL Transaction Output

USER-ID	STATUS	TYPE	OPCODE	RESPONSE	FLAG1
0149	ACTIVE	SCF	TERMINATE	TERMINATE	LAST ENTRY (W)
0000040T	INACT	PLEX	CMD PROC	NORM COMP	ERROR TASK (NS)
			I/O COMP	CHG CMD FILE	TERMINATE
			NEW USER	CHG DATA FILE	PURGE OUT
			PURGE OUT	CHG MSG FILE	
			GOT SPACE	CHG RPT FILE	
			LAST ENTRY	TERM TP USER	
				READ CMD FILE	
				READ DATA FILE	
				WRT MSG FILE	
				WRT RPT FILE	
				GET SPACE	
				LAST ENTRY	

USER-ID

is a unique ID that identifies the user to SYSTEM 2000. This USER-ID is used when terminating a user who enters a TPLX or a TSCF transaction. For terminal-oriented transactions, the USER-ID is the terminal ID. For non-terminal transactions, use the CICS task number followed by the character **T**.

STATUS

indicates whether the user is currently active or inactive in the interface.

TYPE

identifies the type of user: PLEX or SCF.

OPCODE

is the command sent to Multi-User.

RESPONSE

is the Multi-User response to the command that is sent.

FLAG1

is the flag that is used for problem determination.

The LPLX transaction lists the status of all PLEX users or a specific PLEX user. The LSCF transaction lists the status of all SCF users or a specific SCF user. Output for these transactions is identical to the LALL transaction output for the appropriate type of user. The keyword USERID is optional for all three transactions.

TALL, TPLX, and TSCF

Terminate users in the interface.

Syntax

TALL USERID=ALL | *userid*

TPLX USERID=ALL | *userid*

TSCF USERID=ALL | *userid*

Details

The TALL transaction terminates all SCF and PLEX users that currently use interface resources. The TPLX transaction terminates all PLEX users or a specific PLEX user. The TSCF transaction terminates all SCF users or a specific SCF user.

All three transactions require the keyword USERID= followed by its value, and they are useful for freeing resources.

Note: These transactions should be used with caution because resources from both the interface and from Multi-User are freed. For all termination transactions, if the specified user is in an active CICS task at the time a termination request is processed, the task is purged by means of the EXEC CICS SET TASK()FORCEPURGE command. △

UDMP

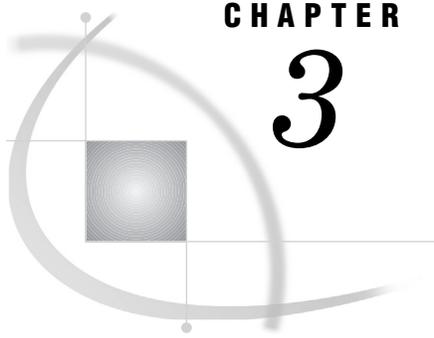
Produces a CICS transaction dump of S2KCUSE for problem analysis.

Syntax

UDMP

Details

Some situations may require help from SYSTEM 2000 Technical Support to resolve a problem. Information critical to the interface is contained in the module S2KCUSE. Transaction UDMP produces a CICS transaction dump of S2KCUSE that has a dump code of UDMP. This transaction has no input parameters.

**CHAPTER****3****PLEX Processing**

<i>Introduction</i>	15
<i>Precompiler Options</i>	15
<i>Command Directives</i>	16
<i>Parameter Execution Options</i>	16
<i>STOP S2K Processing</i>	16
<i>Pseudo-Conversational Transactions</i>	16
<i>COBOL Differences</i>	17

Introduction

The SYSTEM 2000 interface to CICS enables you to write PLEX programs that execute as a CICS program. Coding a CICS PLEX program is similar to coding a batch PLEX program, but you must consider

- precompiler options
- STOP S2K processing
- pseudo-conversational transactions
- COBOL differences.

This chapter contains information you need when writing a CICS PLEX program. For detailed information to help you write a PLEX program, consult the *SYSTEM 2000 R12.1 PLEX Manual*.

Precompiler Options

The PLEX precompiler must be told in what environment the PLEX program will execute and in what language it is written. SYSTEM 2000 software supports COBOL, PL/I, and Assembler. You can use command directives or parameter execution options to provide the precompiler with this information.

Command Directives

\$CICS ON

tells the PLEX precompiler to generate command-level code suitable for CICS processing. This directive can be anywhere within your program.

\$CICS OFF

the directive \$CICS OFF is no longer required but is accepted for compatibility.

Note: The command directive FOR is not supported in Version 1 of SYSTEM 2000. To check PLEX return codes, you must use the \$FOR and \$CFOR command directives or code your own IF statement. For more information about these command directives, see the *SYSTEM 2000 PLEX Manual*. Δ

Parameter Execution Options

LANG=COB2

indicates to the precompiler that the source code is COBOL II. With COBOL II, the precompiler can take advantage of the dynamic call feature and generate dynamic calls to the interface. Only static calls are supported when using COBOL/VS; therefore, a CICS PLEX program could not call the interface directly. If COB2 is not specified, then the precompiler assumes COBOL/VS, and less efficient code is generated.

Note: Unless otherwise specified, any reference to COBOL II also includes COBOL for OS/390. Please read the section “COBOL Differences” on page 17 for an explanation of differences between COBOL versions. Δ

CICS

functions identically to the directive \$CICS ON. Example:

```
//PRECOM EXEC . . . , PARM='LANG=COB2,CICS'
```

STOP S2K Processing

SYSTEM 2000 control blocks are acquired when a PLEX program issues a START S2K command. These control blocks are freed when a STOP S2K command is issued. In order to prevent a shortage of control blocks, SYSTEM 2000 issues a STOP S2K for those programs that do not issue their own STOP S2K.

SYSTEM 2000 does not automatically invoke STOP S2K processing if the S2KCUSE option DOSTOP=N or if the transaction is pseudo-conversational. SYSTEM 2000 uses a CICS task-related user exit to perform automatic STOP S2K processing. Using this exit incurs additional overhead that can be avoided if the program performs its own STOP S2K processing.

Pseudo-Conversational Transactions

Pseudo-conversational transactions are a series of individual transactions grouped together to form a logical transaction. Resources are maintained from transaction-to-transaction to facilitate quick, efficient processing. Normally, it is the responsibility of the last transaction in the series to free these resources upon completion.

For SYSTEM 2000 transactions, a STOP S2K command should be issued to free resources and signal the end of the logical transaction. To prevent the interface from issuing an AUTOMATIC STOP S2K at task termination, it must be aware that the transaction is pseudo-conversational. You do this by

- informing SYSTEM 2000 by means of a START S2K USE(2) command. USE(2) means it is a pseudo-conversational request and that automatic STOP S2K processing is not to occur. This is a more efficient method and is recommended.
- issuing the TRANSID option in the EXEC CICS RETURN command at task termination. This is a less efficient method because a global task-related user exit must then be used to analyze all transactions at task termination.

Note: The S2KUSE options must be DOSTOP=Y and CHKTRNS=Y. △

Each transaction in the series must preserve resources, such as SYSTEM 2000 control blocks, for the next transaction in the series. Each transaction can then restore the required resources and begin its processing in an efficient manner.

By maintaining control blocks S2KDUM, SCHEMA, and COMMBLOCK, fewer resources are consumed both in CICS and in the Multi-User environment. Processing-time improves because where-clause operations are not repeated and stack positions are not rebuilt. The following control blocks can be stored in temporary storage or in a COMMAREA:

S2KDUM

contains the address of the CICS URB and other information set at initialization time that must be preserved throughout the session. Your application program must ensure that S2KDUM is saved and restored for each CICS transaction. The CICS interface needs the information in S2KDUM to associate a re-starting CICS transaction with an already-active user. This may be the most important control block.

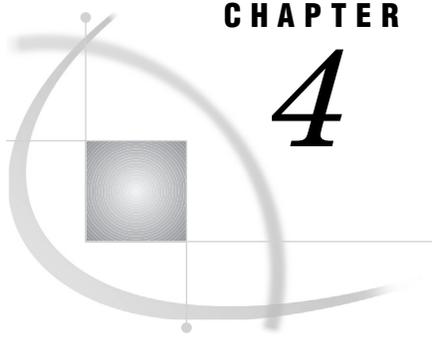
COMMBLOCKS and SCHEMAS

contain internal information, such as the internal database number and component numbers, which are resolved the first time these blocks are used. If the program does not save and restore these control blocks, SYSTEM 2000 must go through the overhead of re-initializing this information for each transaction.

COBOL Differences

You must be aware of the following program invocation differences when you code a CICS PLEX program in COBOL II or COBOL/VS.

- COBOL II supports the dynamic call feature, which allows a program to invoke another program by using a COBOL-type CALL statement that functions like the EXEC CICS LINK command. This means COBOL II programs can access the CICS interface directly and eliminate the extra CALL (to PLXPBLD) necessary for COBOL/VS programs.
- COBOL/VS supports only static calls, which require that the invoked program be link-edited with the calling program. In order to avoid large load modules, SYSTEM 2000 provides the Assembler stub program PLXPBLD to serve as the link between the PLEX program and the interface. PLXPBLD must be link-edited with every CICS COBOL/VS PLEX program that accesses SYSTEM 2000.

**CHAPTER****4****Error Recovery and Security**

<i>Introduction</i>	19
<i>Error Recovery</i>	19
<i>Program Abends</i>	19
<i>Terminal Errors</i>	20
<i>System Abends</i>	20
<i>Shutdown Logic</i>	20
<i>Event Logging Feature in Interface to CICS</i>	21
<i>Logging Record Formats</i>	21
<i>Transaction S2KL</i>	24
<i>S2KL Transaction Messages</i>	25

Introduction

This chapter describes interface provisions for error recovery and shutdown logic.

Error Recovery

CICS provides several user-replaceable programs that are invoked at specific points in CICS processing to facilitate user error recovery. SYSTEM 2000 provides logic to be incorporated into these programs in the following instances:

- program abends (DFHPEP)
- terminal errors (DFHZNEP)
- system abends (DFHSRT).

Program Abends

The CICS program DFHPEP provides a user error recovery point for program abends such as ASRAs. When the interface is installed, DFHPEP should be updated to include the SYSTEM 2000 error recovery logic. If an abend occurs while processing a SYSTEM 2000 transaction, DFHPEP invokes error recovery logic in the interface.

Terminal Errors

The CICS program DFHZNEP provides a user error recovery point for VTAM terminal errors. When the interface is installed, DFHZNEP should be updated to include the SYSTEM 2000 error recovery logic. If a terminal error occurs, DFHZNEP determines whether a SYSTEM 2000 transaction last used the terminal and, if it has, invokes DFHZNEP error recovery logic in the interface.

Many sites now take advantage of the IBM MRO feature, which allows multiple CICS regions to communicate with each other. A typical scenario would be a terminal owning region (TOR) and an application owning region (AOR).

CICS does not provide for error recovery in the AOR when a terminal error occurs in the TOR. However, the SYSTEM 2000 DFHZNEP error recovery logic invokes error recovery logic in the interface of the AOR when a terminal error occurs in the TOR. Some setup is required and additional overhead is incurred to execute this feature. Consult the *SYSTEM 2000 Installation Instructions* for a detailed explanation.

System Abends

The system recovery table (DFHSRT) is provided by CICS to intercept system-type abends. For sites using XMS, the DFHSRT is updated when the interface is installed to include entries for recovering from various system-type abends that are associated with the use of XMS processing.

Although CICS traps the system abend, the PLEX transaction that caused the abend receives an ASRB abend. Subsequent PLEX users receive notification that Multi-User is not active. If the ASRB abend is not an acceptable result, an alternative solution, at a performance cost, can be obtained by contacting your SYSTEM 2000 Technical Support Representative.

For guidelines on XMS, consult your *SYSTEM 2000 Installation Instructions* or contact your SYSTEM 2000 Technical Support Representative.

Shutdown Logic

To ensure that resources acquired by the Multi-User interface on behalf of CICS are freed when CICS terminates, SYSTEM 2000 provides an exit at CICS shutdown time in the form of a Task-Related User Exit. The exit PLXTRUE replaces the functionality that was provided by the PLT program PLXSHUT. PLXTRUE will execute at shutdown time for the following types of shutdown:

Normal or Orderly

Existing CICS tasks are allowed to finish on normal shutdowns. PLXTRUE assumes that active SYSTEM 2000 tasks will free resources upon completion and will not attempt to clean up any associated resources in the Multi-User interface. As a consequence, those transactions that do not free resources leave resources in the Multi-User interface that must now be manually freed.

Immediate

Existing CICS tasks are not allowed to finish on immediate shutdowns. PLXTRUE assumes that all SYSTEM 2000 activity is finished and notifies Multi-User software to free all resources for this CICS region.

Event Logging Feature in Interface to CICS

The new Version 1 feature known as “CICS logging” was developed to aid in SYSTEM 2000 problem resolution. CICS Transient Data Services logs various types of diagnostic information. The following types of log records are written:

Initialization records

contain information that describes both the interface environment and the CICS environment. The first CICS PLEX/SCF transaction that is executed starts the logging feature and causes Initialization format records to be logged.

Abend records

contain information that is used to identify problem programs and terminals. Program abends, terminal errors, and user-requested aborts (TALL, TPLX, TSCF) cause Abend format records to be logged.

Shutdown records

contain information such as PLEX and SCF transaction counts and counts for various error conditions. At CICS shutdown time, Shutdown format records are logged.

Detailed records

are written whenever a condition occurs that is considered out of the ordinary. Although they are not necessarily error conditions, the following situations might be potential problem areas and are, therefore, logged:

- Resource unavailable – no URBs, PCI blocks, and so forth.
- Automatic STOP S2K generated – whenever the interface generates a STOP S2K on behalf of the user.
- PLEX return codes 42 through 90 – might occur due to erroneous user program code.

The first three types of logging occur automatically, however, Detailed logging is optional and can be turned on or off, at your discretion, by using transaction S2KL. Detailed logging is useful for providing additional information when a problem cannot be resolved by using automatic logging alone. Detailed logging may also be helpful in determining whether interface resources are adequate.

Logging Record Formats

Initialization records provide information about the interface and the CICS system settings. Initialization logging occurs only one time during a CICS session. An explanation of the fields shown in Output 4.1 follows the example log.

Output 4.1 Initialization Records Format

```

1           2           3           CICS/SYSTEM 2000 INITIALIZATION SETTINGS

CICS: JOBNAME= 4           RELEASE= 5           MAX TASKS= 6           EXT SEC= 7           MRO= 8
      STARTUP= 9           COMMAND PROT= 10        STORAGE PROT= 11        TRANS ISOLATION= 12

S2K: MULTI-USER= 13        VERSION= 14           COMMUNICATION= 15        DFHZNEP= 16        DFHPEP= 17

PLEX: PLXFRMT= 18        URBS= 19           PWAITS= 20        PLXUEX1= 21        PLXUEX2= 22        PLXUEX3= 23
      PLXTRUE= 24        DOSTOP= 25        CHKTRNS= 26        TCTUA= 27

SCF: SCFDRVR= 28        URBS= 29           PCI= 30        SCFUEX1= 31        SCFUEX2= 32        SCFUEX3= 33        SCFUEX4= 34

```

- 1** message ID S2KLOGI
- 2** date and time record is logged
- 3** VTAM APPLID of CICS region
- 4** CICS job name
- 5** CICS release
- 6** CICS max tasks
- 7** external security is active, Y or N
- 8** CICS has any active MRO connections
- 9** CICS startup date
- 10** CICS command protection is active, Y or N
- 11** CICS storage protection is active, Y or N
- 12** transaction isolation is active, Y or N
- 13** job name of Multi-User region that CICS is communicating with
- 14** version/release of Multi-User
- 15** type of communication between CICS and Multi-User, XMS, or SVC
- 16** version number of DFHZNEP if SYSTEM 2000 code is present, otherwise, NON-S2K
- 17** version number of DFHPEP if SYSTEM 2000 code is present, otherwise, NON-S2K
- 18** version number of PLEX driver program PLXFRMT
- 19** number of PLEX URBS defined
- 20** number of 1-second waits the interface issues for a new PLEX user
- 21** PLEX user exit 1 exists, Y or N
- 22** PLEX user exit 2 exists, Y or N
- 23** PLEX user exit 3 exists, Y or N
- 24** version number for automatic STOP S2K program PLXTRUE
- 25** automatic STOP S2K processing is activated, Y or N
- 26** PLXTRUE considers transactions that issue RETURN with TRANSID as pseudo-conversational, Y or N
- 27** terminal error recovery code for MRO environments is active, Y or N
- 28** version of SCF driver program SCFDRVR
- 29** number of SCF URBS defined
- 30** number of SCF paging control blocks defined
- 31** SCF user exit 1 is present, Y or N

- 32 SCF user exit 2 is present, Y or N
- 33 SCF user exit 3 is present, Y or N
- 34 SCF user exit 4 is present, Y or N.

Abend and Detailed records have identical formats and are distinguished from one another by MSG ID. An explanation of the fields in Output 4.2 follows the example log.

Output 4.2 Abend and Detailed Records Format

MSG ID	DATE	TIME	PROGRAM	TERM	TRANS	USERID	DESCRIPTION
1	2	3	4	5	6	7	8

- 1 message ID, S2KLOGA for Abend type records and S2KLOGD for Detailed type records
- 2 date record was logged
- 3 time record was logged
- 4 SYSTEM 2000 program that logged record
- 5 terminal ID or "none" for non-terminal oriented tasks
- 6 transaction ID
- 7 user ID associated with the terminal or task number for non-terminal oriented tasks
- 8 description of event that caused logging.

Output 4.3 Examples of Abend Records

```
S2KLOGA 11/04/1999 11:30:04 DFHZNEP 0447 S2KU SYSOPER
TERMINAL ERROR, ERROR CODE=X'AF'
```

```
S2KLOGA 11/09/1999 09:18:50 S2KTERM 0284 TSCF SYSOPER
TSCFUSERID=0284
S2KLOGA 11/09/1999 09:18:50 S2KTERM 0284 S2KU SYSOPER
USER ABORTED BY REQUEST
```

Output 4.4 Examples of Detailed Records

```
S2KLOGD 11/04/1999 11:29:39 PLXTRUE NONE EMPV 0000051T
PROGRAM=EMPLOYEV,TASK #=0000051,AUTOMATIC STOP S2K
```

```
S2KLOGD 11/04/1999 11:31:13 SCFDRVR 0456 S2KU SYSOPER
NO STACE BLOCKS AVAILABLE
```

Shutdown Records occur only one time, and that is at CICS shutdown time. Various totals for PLEX and SCF are logged. An explanation of the fields shown in Output 4.5 follows the example log.

Output 4.5 Shutdown Records Format

```

  1      2 CICS/SYSTEM 2000 SHUTDOWN STATISTICS- 3
  PLEX: TRXS= 4 STOP S2KS= 5
        RC42= 6 RC43= 7 RC44= 8 RC70= 9 RC71= 10 RC76= 11 RC77= 12
        RC78= 13 RC79= 14 RC80= 15 RC82= 16 RC83= 17 RC84= 18 RC86= 19
        RC88= 20 RC90= 21

        SCF: TRXS= 22 NO URBS= 23 NO PCI= 24 NO STACE= 25

  ABORT: DFHZNEP= 26 DFHPEP= 27 MRO= 28 USER= 29

```

- 1 message ID of the log record, S2KLOGS
- 2 date and time of shutdown
- 3 type of shutdown, controlled or immediate
- 4 number of CICS transactions that executed PLEX commands
- 5 number of automatic STOP S2Ks issued by PLXTRUE
- 6-21 number of times that PLEX return code occurred
- 22 number of SCF sessions, one for each session that begins with the USER command and ends with the EXIT command
- 23 number of times no SCF URBS were available
- 24 number of times no SCF PCI blocks were available
- 25 number of times no SCF STACE blocks were available
- 26 number of times a terminal error resulted in DFHZNEP requesting cleanup for a PLEX/SCF user
- 27 number of times an abend resulted in DFHPEP requesting cleanup for a PLEX/SCF user
- 28 number of times a terminal error occurred in a TOR region and S2KMRO requested cleanup in the AOR
- 29 number of times a PLEX/SCF user was aborted by user request (TALL, TPLX, or TCF transaction)

Transaction S2KL

Transaction S2KL has two functions: to control Detailed logging and to display Initialization and Shutdown records.

Detailed logging can be turned on or off during CICS execution by executing transaction S2KL with the option DET=.

When you issue S2KLDET=ON or S2KLDET=OFF, you get these messages:

```

S2KLOG - 02 DETAILED LOGGING TURN ON
S2KLOG - 03 DETAILED LOGGING TURNED OFF

```

A terminal display of Initialization logging records and a current snapshot of Shutdown statistics can be obtained by executing transaction S2KL with the DISPLAY option. See the previous section “Logging Record Formats” for a description of the fields.

When you issue S2KLDISPLAY, your output looks like that shown in Output 4.6.

Output 4.6 S2KLDISPLAY Example

```

11/09/1999          SYSTEM 2000/CICS STATISTICS          14:52:03
CICS:JOBNAME=CICSMW41 RELEASE=0410 EXT SEC=Y MRO=Y MAX TASKS=00025
S2K:MULTIUSER=MU2KMWV1 VERS=V1.0-38 COMM=XMS ZNEP=V1.0 PEP=V1.0
PLEX:PLXFRMT=V1.0 URBS=020 WAITS=02 UEX1=N UEX2=N UEX3=N
    PLXTRUE=V1.0 (ENABLED) DOSTOP=Y  CHKTRNS=N TCTUA=N
    TRXS=0001010 STOP S2KS=008
          RC42=000 RC43=000 RC44=000 RC70=000 RC71=000 RC76=000
          RC77=000 RC78=000 RC79=000 RC80=000 RC82=000 RC83=000
          RC84=000 RC86=000 RC88=000 RC90=000
SCF:SCFDRVR=V1.0 URBS=010 PCI=002 UEX1=N UEX2=N UEX3=N UEX4=N
    TRXS=0000902 NO URBS=000 NO PCI=000 NO STACE=000
ABORT:ZNEP=000 PEP=005 MRO=000 USER=003
          PRESS CLEAR KEY TO EXIT

```

S2KL Transaction Messages

S2KLOG - 01 INVALID INPUT, PLEASE CHECK SYNTAX

The parser found a parameter in the input that was unrecognizable or unexpected.

S2KLOG - 02 DETAILED LOGGING TURNED ON

Detailed logging was successfully turned on.

S2KLOG - 03 DETAILED LOGGING TURNED OFF

Detailed logging was successfully turned off.

S2KLOG - 04 UNABLE TO LOAD S2KCUSE

An attempt to load module S2KCUSE was unsuccessful.

S2KLOG - 05 FUNCTIONALITY NOT SUPPORTED FOR THIS RELEASE

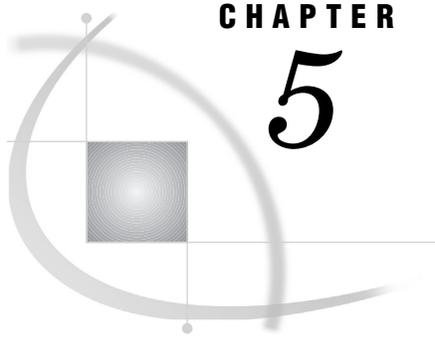
Transaction S2KL is incompatible with this release of CICS.

S2KLOG - 06 ERROR WRITING TO TRANSIENT DATA

An attempt to write to the defined transient data destination was unsuccessful.

S2KLOG - 07 ERROR EXECUTING EXEC CICS COMMAND

One or more CICS commands were unsuccessful during execution.



CHAPTER

5

User Exits

<i>Introduction</i>	27
<i>CICS User Exits</i>	27
<i>Examples</i>	29
<i>Non-CICS User Exits</i>	30

Introduction

This chapter describes user exits provided by the SYSTEM 2000 interface to CICS. The interface supports both user exits that are invoked with the EXEC CICS LINK PROGRAM command and user exits that are called directly from the CICS interface code.

CICS User Exits

The following user exits are invoked by using the EXEC CICS LINK PROGRAM command. The exits must be stand-alone command-level programs. The link to the exits contains a COMMAREA parameter that specifies which control block (SCFCOMM or PLXCOMM) is passed to the exit. The macros that define the common areas are in your SOURCE library. For Assembler language, use the macro PLXCOMA to define the PLXCOMM DSECT. For COBOL and COBOL II, the layout of PLXCOMM is PLXCOMC. For PL/I, the layout is PLXCOMP.

The PLEX common block includes the field PLXURB, which contains the address of the PLEX URB of the user. The layout of the URB can be mapped by using the DURBMU macro. The exits designed for URB alteration can address the URB by using one of the following methods:

- In Assembler language, load a register by using the field DFHEICAP to get the address of the passed COMMAREA. This register can then be used as a base for the PLXCOMM DSECT.
- In COBOL, copy the COBOL definition of PLXCOMM into the linkage section, where it will be initialized to the values of the passed COMMAREA. The name of this area must be set to DFHCOMMAREA in the called-program linkage section.
- In a COBOL II program, the translator generates the DFHCOMMAREA as a 01-level in the linkage section. This gives access to the COMMAREA. You can also define the COMMAREA in working storage, or use a SET ADDRESS command to get addressability to it. The CICS command EXEC CICS ADDRESS COMMAREA obtains the address of the passed common area.
- In PL/I, copy the PL/I definition of PLXCOMM and define the area as a based variable.

Refer to the *CICS Application Programmer's Reference* for more details about accessing passed data.

The available exits and the data areas required are

SCFUEX1

Monitor or modify the input in its unedited form. All terminal dependencies (such as SBAs) are still present in the message. The Paging Control Information (PCI) block is available in this exit for monitoring or modification. Fields relevant to this exit are

SCFCMSGGA	input message address
SCFCMSGGL	input message length
SCFCPCI	PCI block address.

SCFUEX2

Monitor or modify the input message in its edited form. The input message has had all terminal dependencies removed from it and is ready to be passed to Multi-User software. The PCI block is available in this exit for monitoring or modification. Fields relevant to this exit are

SCFCMSGGA	edited message address
SCFCMSGGL	edited message length
SCFCPCI	PCI block address.

SCFUEX3

Journal the edited message. The PCI block is available in this exit for monitoring or modification. Fields relevant to this exit are

SCFCMSGGA	edited message address
SCFCMSGGL	edited message length
SCFCPCI	PCI block address.

SCFUEX4

Monitor or modify the User Request Block (URB). The PCI block and the URB are available in this exit for monitoring or modification. Fields relevant to this exit are

SCFCURB	URB address
SCFCPCI	PCI block address.

PLXUEX1

Monitor or modify the accounting fields of the URB. Link to this exit from the PLXFRMT routine immediately before the call to Multi-User to initialize the user by using the START S2K command. Fields relevant to this exit are

PLXURB	URB address
AS2KDM	S2KDUM address.

PLXUEX2

Monitor the URB and individual SYSTEM 2000 control blocks prior to the call to Multi-User for processing. Fields relevant to this exit are

PLXURB	URB address
AS2KDM	S2KDUM address
ACOMBK	Database commblock address
ASCHEM	Schema address.

PLXUEX3

Monitor the URB and individual SYSTEM 2000 control blocks after the call to Multi-User for processing. Fields relevant to this exit are

PLXURB	URB address
AS2KDM	S2KDUM address
ACOMBK	Database commblock address
ASCHEM	Schema address.

Examples

Method 1:

```

L      R10,DFHEICAP
USING R10,PLXCOMM
L      R11,PLXURB
USING R11,DURB
L      R4,AS2KDM           ADDRESS OF S2KDUM
L      R5,ACOMBK          ADDRESS OF COMMBLOCK
L      R6,ASCHEM          ADDRESS OF SUBSCHEMA
...
PLXCOMA
DURBMU

```

Method 2:

```

L      R10,DFHEICAP
USING R10,PLXCOMM
L      R11,PLXURB
USING R11,DURB
L      R12,URBPARAM
L      R4,0(,R12)          ADDRESS OF S2KDUM
L      R5,4(,R12)          ADDRESS OF COMMBLOCK
L      R6,8(,R12)          ADDRESS OF SUBSCHEMA
...
PLXCOMA
DURBMU

```

For COBOL II or PL/I, you can set pointer variables similar to Method 2. After you have the address of the URB by means of PLXURB, 152 bytes (hex 98) is the address of URBPARAM, which points to a list of the S2KDUM, COMMBLOCK, and SUBSCHEMA addresses.

Other PLXCOMM information you may find useful is about PLXUID, which is an 8-character field that identifies this user. If the transaction originates from a terminal, the PLXUID field contains the 4-character terminal ID followed by 4 blanks. If non-terminal oriented, then the content of PLXUID is a 7-character display number representing the transaction number followed by the letter **T**. All other variables within PLXCOMM are used by the interface and the contents are unpredictable.

Non-CICS User Exits

For optimum performance, if your user exits do not have any CICS commands, you can execute user exits that are called directly from CICS interface code. To invoke these user exits, you must link-edit the exit with the appropriate interface program.

Table 5.1 shows the user exit name, the interface program that invokes it, and the name of the non-CICS user exit.

Table 5.1

User Exit	Program	Non-CICS Exit
SCFUEX1	SCFDRVR	DREXT0
SCFUEX2	SCFDRVR	DREXT1
SCFUEX3	SCFDRVR	DRJRNL
SCFUEX4	SCFINIT	NLIEXT1
PLXUEX1	PLXFRMT	PLIEXT1
PLXUEX2	PLXFRMT	PLREX1
PLXUEX3	PLXFRMT	PLREX2

For SCF, register 1 points to a parameter list that looks like Table 5.2. Register 0 is the address of your URB for NLIEXT1. Register 1 is 0 for DREXT0, DREXT1, and DRJRNL.

Table 5.2

Offset	Parameter List
+0	Address of input message
+4	Length of the input message
+8	Address of the PCI block

For PLEX, register 1 points to a parameter list that looks like Table 5.3. Register 0 points to the address of your URB for all PLEX exits. Register 1 is 0 for PLIEXT1.

Table 5.3

Offset	Parameter List
+0	Address of S2KDUM
+4	Address of COMMBLOCK
+8	Address of SUBSCHEMA

Index

- A**
- Abend log records 21
 - example 23
 - ACOMBK field
 - PLXUEX2 exit 28
 - PLXUEX3 exit 29
 - AS2KDM field
 - PLXUEX1 exit 28
 - PLXUEX2 exit 28
 - PLXUEX3 exit 29
 - ASCHEM field
 - PLXUEX2 exit 28
 - PLXUEX3 exit 29
 - Assembler language, CICS user exits 27
- B**
- BOTTOM command 10
 - browsing SCF pages 4
- C**
- CHKTRNS= option 2
 - CICS logging 21
 - Abend log records 21, 23
 - Detailed log records 21, 23, 24
 - Initialization log records 21, 24
 - record formats 21
 - S2KL transaction 24
 - S2KL transaction messages 25
 - Shutdown log records 21, 24
 - \$CICS OFF directive 16
 - \$CICS ON directive 16
 - CICS operator communication transaction 12
 - CICS option 16
 - CICS user exits 27
 - Assembler language 27
 - COBOL language 27
 - PL/I language 27
 - PLXCOMA macro 27
 - PLXCOMC macro 27
 - PLXCOMP macro 27
 - PLXUEX1 28
 - PLXUEX2 28
 - PLXUEX3 29
 - SCFUEx1 28
 - SCFUEx2 28
 - SCFUEx3 28
 - SCFUEx4 28
 - CLEAR key, assigning values to 5
 - COBOL language
 - CICS user exits 27
 - PLEX transactions 17
 - command directives 16
 - Command Editor
 - See SCF Command Editor
 - COMMBLOCK control block 17
 - compatibility 2
 - control blocks 16
 - control commands 8
- D**
- DELETE command 11
 - Detailed log records 21, 23, 24
 - DFHPEP program 19
 - DFHSRT program 20
 - DFHZNEP program 20
 - DOSTOP= option 2
 - DOWN command 11
 - dumps 14
- E**
- END command 8
 - error messages, S2KL transaction 25
 - error recovery 19
 - See also CICS logging
 - DFHPEP program 19
 - DFHSRT program 20
 - DFHZNEP program 20
 - event logging 21
 - freeing resources 20
 - PLXSHUT exit 20
 - PLXTRUE exit 20
 - program abends 19
 - shutdown logic 20
 - system abends 20
 - terminal errors 20
 - transaction dumps 2, 14
 - event logging
 - See CICS logging
 - exits
 - See CICS user exits
 - See non-CICS user exits
- F**
- FIND command 11
 - FOR directive 16
- G**
- GET command 9
- I**
- Initialization log records 21, 24
 - Input Area, SCF Command Editor 10
 - INSERT command 11
 - interface, activating 4
- J**
- JCL, submitting from Command Editor 9
- L**
- LALL transaction 13
 - LANG= option 16
 - libraries
 - listing 9
 - SCF Command Editor 8
 - library commands 9
 - library members, editing
 - See SCF Command Editor, library members
 - LISTD command 9
 - logging
 - See CICS logging
 - logging Command Editor commands 11
 - logon screen, bypassing 4
 - LPLX transaction 13
 - LSCF 13
 - LSCF transaction 13
- M**
- member commands 9
 - members, editing
 - See SCF Command Editor, library members

messages, S2KL transaction 25
 MODIFY command 11

N

non-CICS user exits 30

O

operator communication transaction 12

P

P (paging) commands 4
 PA keys, assigning values to 5
 pages, SCF
 See SCF pages
 parameter execution options, PLEX 16
 performance
 non-CICS user exits 30
 STOP S2K processing 2
 PF keys 5
 S2OP environment 6, 12
 SCF Command Editor 5, 6
 PL/I language, CICS user exits 27
 PLEX transactions 15
 \$CICS OFF directive 16
 \$CICS ON directive 16
 CICS option 16
 COBOL differences 17
 command directives 16
 COMMBLOCK control block 17
 FOR directive 16
 freeing resources 17
 LANG= option 16
 operating environment, specifying 15
 parameter execution options 16
 precompiler options 15
 pseudo-conversational transactions 16
 return codes, checking 16
 S2KDUM control block 17
 SCHEMA control block 17
 STOP S2K processing 16
 SYSTEM 2000 control blocks 16
 PLEX users 13
 listing status of 13
 terminating 14
 PLXCOMA macro 27
 PLXCOMC macro 27
 PLXCOMP macro 27
 PLXSHUT exit 20
 PLXTRUE exit 20
 PLXUEX1 exit 28
 PLXUEX2 exit 28
 PLXUEX3 exit 29
 PLXURB field
 PLXUEX1 exit 28
 PLXUEX2 exit 28
 PLXUEX3 exit 29
 precompiler options, PLEX 15
 product-supplied transactions
 See SCF transactions
 program abends 19

program function keys
 See PF keys
 pseudo-conversational transactions 16
 checking for 2

R

resources, freeing
 error shutdown 20
 PLEX transactions 17
 return codes (PLEX), checking 16

S

S2KCUSE module 2
 transaction dump 14
 S2KDUM control block 17
 S2KE command 9
 S2KE transaction 7
 S2KK command 9
 S2KK transaction
 and Command Editor file space 6
 syntax 5
 S2KL transaction 24
 S2KL transaction messages 25
 S2KU transaction 4
 S2OP environment, PF key settings 6, 12
 S2OP transaction 12
 SAVE command 10
 SCF Command Editor
 control commands 8
 ending 8
 finding text strings 11
 initializing 9
 Input Area 10
 invoking 7
 libraries and members 8
 library and member commands 9
 moving the line pointer 11
 PF key settings 5, 6
 S2KK transaction and file space 6
 Status Area 10
 submitting JCL from 9
 Text Display Area 10
 text editor commands 10
 user identification 8
 SCF Command Editor, commands
 BOTTOM 11
 control 8
 DELETE 11
 DOWN 11
 END 8
 FIND 11
 GET 9
 INSERT 11
 library 9
 LISTD 9
 logging 12
 member 9
 MODIFY 11
 S2KE 9
 S2KK 9
 SAVE 10
 SCRATCH 10

SEND 8
 SUBMIT 9
 summary table 7
 text editing 10
 TOP 11
 UP 11
 SCF Command Editor, libraries 8
 listing directories in 9
 SCF Command Editor, library members 8
 loading 9
 removing 10
 saving 10
 SCF Command Editor, lines
 deleting 11
 inserting 11
 modifying 11
 SCF pages 4
 browsing 4
 characters per line, setting 5
 lines per page, setting 5
 output line limit, setting 5
 paging commands 4
 SCF transactions 3
 LALL 13
 LPLX 13
 LSCF 13
 S2KE 7
 S2KK 5
 S2KU 4
 S2OP 12
 TALL 14
 TPLX 14
 TSCF 14
 UDMP 14
 SCF users
 listing status of 13
 terminating 14
 SCFCMSG field
 SCFUEX1 exit 28
 SCFUEX2 exit 28
 SCFUEX3 exit 28
 SCFCMSG field
 SCFUEX1 exit 28
 SCFUEX2 exit 28
 SCFUEX3 exit 28
 SCFCPCI field
 SCFUEX1 exit 28
 SCFUEX2 exit 28
 SCFUEX3 exit 28
 SCFUEX4 exit 28
 SCFCURB field 28
 SCFUEX1 exit 28
 SCFUEX2 exit 28
 SCFUEX3 exit 28
 SCFUEX4 exit 28
 SCHEMA control block 17
 SCRATCH command 10
 security 19
 See also CICS logging
 DFHPEP program 19
 DFHSRT program 20
 DFHZNEP program 20
 event logging 21
 freeing resources 20
 PLXSHUT exit 20
 PLXTRUE exit 20
 program abends 19

- shutdown logic 20
- system abends 20
- terminal errors 20
- transaction dumps 2, 14
- SEND command 8
- Shutdown log records 21, 24
- shutdown logic 20
- sign-on screen, bypassing 4
- Status Area, SCF Command Editor 10
- STOP S2K processing
 - automatic 2
 - PLEX transactions 16
- SUBMIT command 9
- SYSTEM 2000 CICS interface 1
 - activating 4
 - compatibility 2
 - functions of 1

- SYSTEM 2000 control blocks 16
- system abends 20

T

- TALL transaction 14
- terminal errors 20
- Text Display Area, SCF Command Editor 10
- text editing commands 10
- TOP command 11
- TPLX transaction 14
- transaction dumps 14
- transactions
 - See PLEX transactions
 - See SCF transactions

- troubleshooting
 - See error recovery
- TSCF transaction 14

U

- UDMP transaction 14
- UP command 11
- user exits
 - See CICS user exits
 - See non-CICS user exits
- user identification, Command Editor 8
- user-written transactions
 - See PLEX transactions
- users
 - See PLEX users
 - See SCF users

Welcome * Bienvenue * Willkommen * Yohkoso * Bienvenido

SAS Publishing Is Easy to Reach

Visit our Web page located at www.sas.com/pubs

You will find product and service details, including

- **sample chapters**
- **tables of contents**
- **author biographies**
- **book reviews**

Learn about

- **regional user-group conferences**
- **trade-show sites and dates**
- **authoring opportunities**
- **custom textbooks**

Explore all the services that SAS Publishing has to offer!

Your Listserv Subscription Automatically Brings the News to You

Do you want to be among the first to learn about the latest books and services available from SAS Publishing? Subscribe to our listserv **newdocnews-l** and, once each month, you will automatically receive a description of the newest books and which environments or operating systems and SAS® release(s) that each book addresses.

To subscribe,

1. Send an e-mail message to listserv@vm.sas.com.
2. Leave the "Subject" line blank.
3. Use the following text for your message:

subscribe NEWDOCNEWS-L *your-first-name your-last-name*

For example: subscribe NEWDOCNEWS-L John Doe

Create Customized Textbooks Quickly, Easily, and Affordably

SelecText® offers instructors at U.S. colleges and universities a way to create custom textbooks for courses that teach students how to use SAS software.

For more information, see our Web page at www.sas.com/selecttext, or contact our SelecText coordinators by sending e-mail to selecttext@sas.com.

You're Invited to Publish with SAS Institute's User Publishing Program

If you enjoy writing about SAS software and how to use it, the User Publishing Program at SAS Institute offers a variety of publishing options. We are actively recruiting authors to publish books, articles, and sample code. Do you find the idea of writing a book or an article by yourself a little intimidating? Consider writing with a co-author. Keep in mind that you will receive complete editorial and publishing support, access to our users, technical advice and assistance, and competitive royalties. Please contact us for an author packet. E-mail us at sasbbu@sas.com or call 919-677-8000, extension 7921. See the SAS Publishing Web page at www.sas.com/pubs for complete information.

See *Observations*®, Our Online Technical Journal

Feature articles from *Observations*®: *The Technical Journal for SAS*® Software Users are now available online at www.sas.com/obs. Take a look at what your fellow SAS software users and SAS Institute experts have to tell you. You may decide that you, too, have information to share. If you are interested in writing for *Observations*, send e-mail to sasbbu@sas.com or call 919-677-8000, extension 7921.

Book Discount Offered at SAS Public Training Courses!

When you attend one of our SAS Public Training Courses at any of our regional Training Centers in the U.S., you will receive a 15% discount on book orders that you place during the course. Take advantage of this offer at the next course you attend!

SAS Institute
SAS Campus Drive
Cary, NC 27513-2414
Fax 919-677-4444

E-mail: sasbook@sas.com
Web page: www.sas.com/pubs
To order books, call Fulfillment Services at 800-727-3228*
For other SAS business, call 919-677-8000*

* **Note:** Customers outside the U.S. should contact their local SAS office.