

ULSRPSFS - User-Written Service Routine

ULSRPSFS is a user-written service routine that can be used to authorize the use of specific application functions, programs, modules, or Com-plete utility programs.

Note:

Because security afforded by this service routine is system-wide, thorough design and analysis is recommended before implementation in order to ensure minimal impact on performance.

When Com-plete is initially installed, a dummy ULSRPSFS module exists as a member of the distribution source library to serve as a guide to the user. This routine must be coded and assembled and can either be link edited as a routine in the Com-plete control program, or loaded dynamically during initialization. The initial Com-plete control program contains a dummy ULSRPSFS service routine that performs no security checking.

Since ULSRPSFS executes as a service routine of Com-plete, it will be dispatched in supervisor state using the key assigned to Com-plete. Consequently, if an abnormal termination occurs while ULSRPSFS is executing, Com-plete also abends.

Note that because ULSRPSFS is entered frequently, it is not advisable for it to perform any I/O functions. Tables and any referenced subroutines should be made part of the service routine.

This chapter covers the following topics:

- How to Use ULSRPSFS
 - ULSRPSFS Conventions
-

How to Use ULSRPSFS

At entry to ULSRPSFS, a set of parameters is received in the form of fullword addresses pointed to by register 1. Word 1 of the parameter list contains the address of the return code indicating the security status. Word 2 contains the address of the user ID accounting block. This control block is created for each user at successful logon to Com-plete. If a user ID requests a function protected by ULSRPSFS, a call is made to ULSRPSFS.

Words 3 and 4 of the parameter list identify the function requested by a program, and word 6 identifies the program making the request. Since authorization for access to most functions is restricted on a user ID basis rather than on a program basis, it is necessary to also identify the terminal and/or batch user making the request. This is accomplished by passing the required information through words 2 and 5 of the parameter list. Word 7 points to two status bytes; the first indicates whether or not a program has been attached, and the second indicates whether or not an Adabas call exists at the time of entry to the ULSRPSFS.

Security violations are indicated in ULSRPSFS by setting a return code. The physical location of the return code is in the routine that passes control to ULSRPSFS. The address of the return code is passed as word 1 of the parameter list. If the function requested is not to be allowed, the return code must be set to 4; otherwise, the return code must be set to 0. At entry to ULSRPSFS, the return code is always initialized to 0.

ULSRPSFS Conventions

The following summarizes the ULSRPSFS linkage conventions.

Attributes

Reentrant.

Type

Nucleus.

Size

No restrictions.

Registers at Entry

Register 1	Address of the parameter list
Register 13	Address of an 18-fullword save area
Register 14	Return address in the calling module
Register 15	Entry address within ULSRPSFS

Registers at Return

Registers 2 through 13 must be unchanged.

Parameters

Word 1	Address of a return code halfword
Word 2	Address of the user ID accounting block
Word 3	Address of a request-type halfword:

	<p>0 Program call (terminal).</p> <p>4 LOAD request.</p> <p>8 LOADT request.</p> <p>12 FETCH request.</p> <p>16 SDOPEN request.</p> <p>32 ATTACH request, where R3 points to the UPCB. R3+12 contains the address of a parameter list: Parm 1 is the address of the eight-character program name of the program being attached. Parm 2 (optional) is the address of the data area being passed. Parm 3 (optional) is the length of the data area being passed.</p> <p>36 LINK, LOAD, XCTL, COLINK, COLOAD, COXCTL request, where R3 points to the UPCB. R3+8 is R0 at the time of request. R3+12 is R1 at the time of request. R3+68 is RF at the time of request. R3+12 is the address of the user parameter list. The first parameter in the parameter list points to an eight-character program name. For LINK, LOAD, and XCTL; R0, R1, and RF are as expanded by MVS LINK, LOAD, and XCTL.</p> <p>40 Indicates that word 8 points to an eight-byte location containing the error message ID in character format (e.g., ZSR0006).</p> <p>44 Catch-all code; indicates that the parameter list is extended by three additional fullwords (words 9, 10, and 11) used to identify the operation code or SVC being requested that does not fall within the range of other operation codes. Com-plete OP codes are found in member CCTOP in COM.SOURCE. These records are only written if APPLYMOD 3 is specified.</p> <p>100 Logoff.</p>
Word 4	Address of the requested eight-character name padded with blanks. This is the name of the program being loaded, the name of the SD file being opened, etc.
Word 5	Address of a halfword containing the TID. If a batch program, the high-order byte of the halfword is an X'FF'.

Word 6	Address of the requesting program name (job name for batch, except request types 0 and 100).					
Word 7	Address of the two status bytes:					
	<table> <tr> <td>Byte 1</td> <td>A = Attached program</td> <td>sp = Non-attached program</td> </tr> <tr> <td>Byte 2</td> <td>A = Open to Adabas at entry</td> <td>sp = Not open to Adabas</td> </tr> </table>	Byte 1	A = Attached program	sp = Non-attached program	Byte 2	A = Open to Adabas at entry
Byte 1	A = Attached program	sp = Non-attached program				
Byte 2	A = Open to Adabas at entry	sp = Not open to Adabas				
Word 8	Not used.					
Word 9	Address of a one-byte area:					
	X'FF' SVC issued					
	X'00' Com-plete OP issued					
	This entry exists only if word 3 points to a halfword 44 and only if OP trapping is enabled.					
Word A	Address of the halfword SVC or OP. This entry exists only if word 3 points to a halfword 44.					
Word B	Address of the UPCB. This entry exists only if word 3 points to a halfword 44.					

Return Codes

0	Allow the request.
4	Disallow the request.

Considerations

a	Runs as a portion of the Com-plete nucleus; it is there fore advisable to maintain a backup either of this module or of the entire Com-plete control program.
b	Tables and reference routines must be made part of the service routine.
c	I/O must be avoided.
d	MCALL is not permitted.
e	Com-plete functions are not permitted.
f	Overlays cannot be used.
g	GETMAINs must be used for storage.
h	GETMAINs are from the Com-plete region, not thread.
i	Called in Com-plete's key.
j	All control blocks and areas passed for the use of the exit will be in Com-plete's key.
k	Active thread will generally be in a different key to Com-plete's key.