OS/390

UNIX System Services: APARs OW42811 and OW42841

OS/390

UNIX System Services: APARs OW42811 and OW42841

Contents

About This Guide

This document supports APARs OW42811 and OW42841 for OS/390 UNIX System Services (OS/390 UNIX), which are available for OS/390 Version 2 Releases 8 and 9. This document is available only on the OS/390 UNIX web site at:

http://www.s390.ibm.com/oe/release/apar.html

Part 1. APARs OW42811 and OW42841: OS/390 UNIX System Services Library

Chapter 1. APARs OW42811 and OW42841: OS/390 UNIX System Services Planning

1.1 Chapter 12. Summary of Interface Changes

This section summarizes the new and changed interface components of OS/390 UNIX.

1.1.1 Operator Commands

1.1.2 Shell Commands

Figure 1 lists new and changed OS/390 UNIX shell commands. See *OS/390 UNIX System Services Command Reference* for more detailed information about these commands.

Figure 1 (Page 1 of 4). Summary of New and Changed Shell Commands

Command Name	Release	Description	Related Support
automount	V2R7	**New parameters:** SECURITY | NOSECURITY specifies whether security checks are to be enforced for files in the file system.	OS/390 UNIX security
bpxmtext	V2R7	**New command:** Displays the description and action text for a reason code returned from the kernel.	dbx
cancel	V2R5	**Updated:** If you are using the OS/390 Print Server Feature, your system automatically uses that version of the **cancel** command.	OS/390 Print Server
chgrp	V2R5	**Changed command:** With the **-h** option, does not follow the symbolic link but instead makes the changes to the symbolic link itself.	Symbolic link
chmod	V2R5	**Changed command:** With the **-h** option, does not allow permission changes if the file is a symbolic link.	Symbolic link
chmount	V2R9	**New command:** Changes the mount attributes of a file system.	Shared HFS
chown	V2R5	**Changed command:** With the **-h** option, does not follow the symbolic link but instead makes the changes to the symbolic link itself.	Symbolic link
chroot	V2R7	**New command:** Changes the root directory for the execution of a command. The new root directory also contains its children.	Root directory
cp	V2R8	**New function:** Copies files to and from MVS data sets.	MVS data sets
confighfs	V2R7	**New command:** Lets interactive shell users query HFS limits. Superusers can set maximum and minimum limits for virtual storage.	Virtual storage
configstk	V2R3	**New command:** Configures the AF_UEINT stack.	AF_UEINT

Figure 1 (Page 2 of 4). Summary of New and Changed Shell Commands

Command Name	Release	Description	Related Support
dbx	V2R6	**Changed subcommand:** The register subcommand has a new register name, $frbN, and new flags, $noflbregs and $flprecision.	IEEE floating point register
	V2R7	**New subcommands:** The set subcommand defines a value for the debug program variable. The onload subcommand defers the building of stop or trace events until the procedure or sourceline is defined in the program that is being debugged.	Debugger
	V2R9	**New subcommand:** The readwritelock subcommand displays the read/write lock information.	Shared HFS
	V2R7	**New function:** Long long variables can be used in expressions, unsigned long long can be used in casting operations, and the examine storage subcommand supports the ld, lo, and lx modes to display data.	Long long variable
df	V2R9	**New fields:** Displays new fields for the system ID of the mounted file system server and the system ID that issued the quiesce request.	Shared HFS
extattr	V2R4	**New command:** Sets, resets, and displays extended attributes for executable HFS files and also allows HFS files to run APF-authorized or program-controlled.	Extended attributes
	V2R9	**New extended attribute:** Enables load modules to be loaded from the shared library region.	Shared library
filecache	V2R4	**New command:** Manages the kernel file cache for files that are read-only; it requires superuser authority.	File cache
fuser	V2R9	**New command:** Lists the process ID of all processes that have one or more open files.	UNIX98
ipcs	V2R9	**New output:** The **-x** output shows information associated with message queue serialization using the perform lock operation (PLO) instruction.	Message queue serialization
link	V2R7	**New command:** Creates a hard link to an existing file.	UNIX 98
lp	V2R5	**Updated function:** If you are using the OS/390 Print Server Feature, your system automatically uses that version of the **lp** command.	OS/390 Print Server
lpstat	V2R5	**Updated function:** If you are using the OS/390 Print Server Feature, your system automatically uses that version of the **lpstat** command.	OS/390 Print Server

Figure 1 (Page 3 of 4). Summary of New and Changed Shell Commands

Command Name	Release	Description	Related Support
ls	V2R4	**New option:** The **-E** option displays extended attributes for regular files.	Extended attributes
	V2R9	**Changed function:** Because of changes to enable a read-only file system, users may need to specify the **-L** option when listing the contents of a directory that is newly defined as a symbolic link to a directory. If you want to add an alias to the **ls** command, see ..	Shared HFS
	V2R9	**Changed option:** The **-E** option indicates whether the program was loaded from the shared library region.	Shared library
man	V2R7	**Updated command:** Provides help information about OS/390 UNIX TSO/E commands.	TSO/E commands
mount	V2R9	**New command:** Logically mounts a file system.	Shared HFS
mv	V2R8	**New function:** Moves files to and from MVS data sets.	MVS data sets
od	V2R6	**New option: F** specifies that the command is to interpret floating-point numbers as being in IEEE format.	IEEE floating point support
passwd	V2R5	**New command:** Changes the login password for the specified userid.	OS/390 UNIX security
pax	V2R7	**Changed function:** Preserves external links and extended attributes by default, and supports extended file attributes.	File system support
	V2R8	**Changed function:** Reads, writes, and lists archive files that are MVS data sets.	MVS data sets
	V2R9	**Changed function:** Supports link names that are longer than 100 characters.	Long link names
poe (and other Parallel Environment commands)	V2R4	**New commands:** Depends on pmd for V2R4. See *OS/390 UNIX System Services Parallel Environment: Operation and Use* V2R4.	Parallel Environment
	V2R7	**New/changed commands:** Depends on pmd for V2R7. See *OS/390 UNIX System Services Parallel Environment: Operation and Use* V2R7 for new options and functionality.	
	V2R9	**New/changed commands:** Depends on pmd for V2R9. See *OS/390 UNIX System Services Parallel Environment: Operation and Use* V2R9 for new options and functionality.	
printenv	V1R2	**New command:** Displays the values of environment variables.	Environment variables
sh	V2R8	**New function:** Use reserved-word commands to create compound commands.	Compound commands
skulker	V2R10	**New command**: Removes old files from a directory based on the date the file was last accessed.	RAS Enhancements
su	V2R8	**New option:** The **-s** option does not prompt for the password.	Surrogate userids

Figure 1 (Page 4 of 4). Summary of New and Changed Shell Commands

Command Name	Release	Description	Related Support
sysvar	V2R9	**New command:** Obtains substitution text for system variables that are defined in IEASYMxx or in the system IPL parameters.	System variables
tar	V2R7	**Changed function:** Preserves external links and extended attributes by default and supports extended file attributes.	File system
	V2R8	**Changed function:** Reads, writes, and lists archive files that are MVS data sets.	MVS data sets
	V2R9	**Changed function:** Supports link names that are longer than 100 characters.	Long link names
	V2R9	**Changed function:** Saves and restores external attribute when using the USTAR (**U**) command.	External attributes
unlink	V2R7	**New command:** Removes a directory entry.	UNIX 98
unmount	V2R9	**New command:** Removes a file system from the file hierarchy.	Shared HFS
uucc	V1R2	**Updated:** Reads the contents of the UUCP configuration files and compiles them into a single configuration file.	UUCP
uconvdef	V2R3	**New command:** Creates binary conversion tables.	Binary conversion tables
uucp	V2R1	**New command:** Copies files between remote UUCP systems.	UUCP
uulog	V1R2	**New command:** Displays information about UUCP events, such as file transfers and remote command execution.	UUCP
uuname	V1R2	**New command:** Displays a list of all remote systems known to UUCP.	UUCP
uupick	V1R2	**New command:** Manages files in the UUCP public receive directory that were sent to you via the **uucp** command.	UUCP
uustat	V1R2	**Updated:** Displays status of pending UUCP transfers.	UUCP
uuto	V1R2	**Updated:** Copies files to users on another system.	UUCP
uux	V1R2	**Updated:** Specifies that a certain command be executed on another site.	UUCP
wall	V2R5	**New command:** Sends a message to all logged-in users.	None
whoami	V2R5	**New command:** Displays a user name associated with the effective userid.	None

1.1.3 Chapter 14. Customizing OS/390 UNIX

1.1.3.1 Customizing the BPXPRMxx Parmlib Members

1.1.3.1.1 Defining File Systems

1.1.3.1.2 Defining System Limits: You can customize your BPXPRMxx parmlib member to provide the performance needed for the way your installation uses kernel services.

CTRACE: Use CTRACE statement to provide tracing while the kernel is starting and to avoid having to issue a TRACE operator command to set tracing options. See . for information about specifying your customized component trace parmlib members.

The only way to change any CTRACE value is with the TRACE command. You cannot use the SETOMVS or SET OMVS command to change the value.

MAXASSIZE: MAXASSIZE is the maximum region size (in bytes) for an address space. You can set a system-wide limit in BPXPRMxx and then set higher limits for individual users. Use the RACF ADDUSER or ALTUSER command to specify the ASSIZEMAX limit on a per-user basis as follows:

```
ALTUSER userid OMVS(ASSIZEMAX(nnnn)
```

MAXCPUTIME: MAXCPUTIME is the time limit (in seconds) for processes that were created by **rlogind** and other daemons. You can set a system-wide limit in BPXPRMxx and then set higher limits for individual users. Use the RACF ADDUSER or ALTUSER command to specify the CPUTIMEMAX limit on a per user basis as follows:

```
ALTUSER userid OMVS(CPUTIMEMAX(nnnn))
```

MAXFILEPROC: Use MAXFILEPROC to determine the number of character-special files, **/dev/fd***xx*, that a single process can have open concurrently. You can also limit the amount of system resources available to a single user process.

When selecting a value, consider the following factors:

- For conformance to standards, set MAXFILEPROC to at least 16 to conform to the POSIX standard or at least 25 to conform to the FIPS standard.

  It is recommended that you set this value to 256.

- The minimum value of 3 supports stdin, stdout, and stderr.
- The value must be larger than 3 to support shell users. If the value is too small, the shell may issue the message “File descriptor not available.” If this message occurs, increase the MAXFILEPROC value.

A process can change the MAXFILEPROC value using the setrlimit() function. Only processes with appropriate privileges can increase their limits.

You can set a system-wide limit in BPXPRMxx and then set higher limits for individual users. Use the RACF ADDUSER or ALTUSER command to specify the FILEPROCMAX limit on a per user basis as follows:

```
ALTUSER userid OMVS(FILEPROCMAX(nnnn))
```

1.1.4.4.1, “Dynamically Changing Certain BPXPRMxx Parameter Values” on page 20 explains how to dynamically change the MAXFILEPROC value.

MAXMMAPAREA: For MAXMMAPAREA, you can set a system-wide limit in BPXPRMxx and then set higher limits for individual users. Use the RACF ADDUSER or ALTUSER command to specify the MMAPAREAMAX limit on a per user basis as follows:

```
ALTUSER userid OMVS(MMAPAREAMAX(nnnn))
```

MAXPROCSYS: You can manage system resources by limiting the number of processes that the system is to support. The values that you specify for MAXPROCSYS, MAXPROCUSER, and MAXUIDS are interrelated. When selecting a value for MAXPROCSYS, remember that these processes are needed:

- The initialization process (BPXOINIT)
- **/usr/sbin/init**, for starting and processing
- **exec sh** to run a shell script
- The process in which the shell script runs

Plan on one process for each daemon (for example, **inetd** and **cron**) that you start from a shell script such as **/etc/rc**. In addition, each shell user needs a minimum of three processes and possibly a few more for piping between shell commands.

Do not specify a higher value for MAXPROCSYS than your system can support because most processes use an entire MVS address space. This value will vary, depending on your environment. If you set the value too high, failures (EAGAIN) for fork or spawn might occur because WLM could not provide enough fork initiators.

1.1.4.4.1, “Dynamically Changing Certain BPXPRMxx Parameter Values” on page 20 explains how to dynamically change the MAXPROCYS value.

For an example of MAXPROCSYS, MAXPROCUSER, MAXRTYS, MAXPTYS, and MAXUIDS settings in BPXPRMxx, see ..

MAXPROCUSER: To improve performance, use MAXPROCUSER to limit user activity. For a typical shell user who starts up 1 to 3 shells, set the limit to 10.

When selecting a value, consider the following factors:

- Set MAXPROCUSER to at least 16 to conform to the POSIX standard for CHILD_MAX, or to at least 25 to conform to the FIPS standard.
- A low MAXPROCUSER value limits the number of concurrent processes that a user can run. A low value limits a user's consumption of processing time, virtual storage, and other system resources.
- Some daemons or users run without UID(0), and may create many address spaces. In these cases, give the daemon ID a high enough PROCUSERMAX value in the OMVS segment.

A user with a UID of 0 is not limited by the MAXPROCUSER value because a superuser may need to be able to log on and use kernel services to solve a problem.

Though not recommended, the security administrator can give the same OMVS UID to more than one TSO/E user ID. Therefore, the number of users can be greater than the number of UIDs that are defined. Check with the security administrator; if users share UIDs, you will need to define a greater number of processes for each user.

You can set a system-wide limit in BPXPRMxx and then set higher limits for individual users. Use the RACF ADDUSER or ALTUSER command to specify the PROCUSERMAX limit on a per-user basis as follows:

```
ALTUSER userid OMVS(PROCUSERMAX(nnnn))
```

MAXPTYS: Use MAXPTYS to manage the number of interactive shell sessions, where each interactive session requires one pseudo-TTY pair. Do not specify an arbitrarily high value for MAXPTYS. But, because each user may have more than one session, it is recommended that you allow four pseudo-TTY pairs for each user (MAXUIDS * 4). Specify a MAXPTYS value that is at least twice the MAXUIDS value.

1.1.4.4.1, "Dynamically Changing Certain BPXPRMxx Parameter Values" on page 20 explains how to dynamically change the MAXPTYS value. For more information about pseudoterminal files, see ..

MAXRTYS: MAXRTYS enables you to manage the number of interactive shell sessions that are accessed by Communications Server terminal support. When you specify this value, each interactive session requires one remote TTY. Avoid specifying an arbitrarily high value for MAXRTYS. However, because each user may have more than one session, you should allow four remote TTY files for each user (MAXUIDS * 4).

The MAXRTYS value influences the configuration of Communications Server nodes and associated terminal files. For more information, see *OS/390 UNIX System Services Communications Server Guide*.

1.1.4.4.1, "Dynamically Changing Certain BPXPRMxx Parameter Values" on page 20 explains how to dynamically change the MAXPROCYS value.

MAXTHREADS: MAXTHREADS is the maximum number of threads that a single process can have active concurrently. If an application needs to create more than the recommended maximum in SAMPLIB, it must minimize storage allocated below the 16M line by specifying C run-time options. For information on the set_thread_limit service (BPX1STL), refer to *OS/390 UNIX System Services Programming: Assembler Callable Services Reference*.

You can set a system-wide limit in BPXPRMxx and then set higher limits for individual users by using the RACF ADDUSER or ALTUSER command to specify the THREADSMAX limit on a per user basis as follows:

```
ALTUSER userid OMVS(THREADSMAX(nnnn))
```

MAXTHREADTASKS: MAXTHREADTASKS is the maximum number of MVS tasks that a single process can have concurrently active.

A high MAXTHREADTASKS value may affect storage and performance. Each task requires additional storage for the following:

- The control blocks built by the kernel
- The control blocks and data areas required by the run-time library

- System control blocks such as the TCB and RB

MAXUIDS: MAXUIDS limits the number of active UIDs. When you select a value for MAXUIDS, consider the following factors:

- Because users are likely to run with three or more concurrent processes each, they require more system resources than typical TSO/E users.
- If the MAXUIDS value is too high relative to the MAXPROCSYS value, too many users can invoke the shell. All users may be affected, because forks might begin to fail.

  For example, if your installation can support 400 concurrent processes—MAXPROCSYS(400)—and each UID needs an average of 4 processes, then the system can support 100 users. For this operating system, specify MAXUIDS(100).

PRIORITYGOAL: If you are using your system to run a critical real-time application program, set the performance groups or service classes to meet the needs of the application program. It is difficult to run both real-time application programs and general users on the same OS/390 UNIX system. There is no mechanism to restrict any set of users from access to the nice() and setpriority() functions. For more information, see ..

PRIORITYPG: If you are using your system to run a critical real-time application program, set the performance groups or service classes to meet the needs of the application program. It is difficult to run both real-time application programs and general users on the same OS/390 UNIX system. There is no mechanism to restrict any set of users from access to the nice() and setpriority() functions. For more information, see .

STEPLIBLIST: With STEPLIBLIST, programs can have temporary access to files that are not normally accessible to other users. Step libraries have many uses; one is so that selected users can test new versions of run-time libraries before the new versions are made available to everyone on the system. Customers who do not put the Language Environment run-time library SCEERUN into the linklist should put the SCEERUN data set name in this file.

If your installation runs programs that have the setuid or setgid bit turned on, only those load libraries that are found in the STEPLIBLIST sanction list are set up as step libraries in the environment that those programs will run in. Because programs with the setuid or setgid bit turned on are considered privileged programs, they must run in a controlled environment. The STEPLIBLIST sanction list provides this control by allowing those programs to use only the step libraries that are considered trusted by the installation.

IBM recommends that the pathname of the file be **/etc/steplib**. This fits in with the IBM strategy to place all customized data in the **/etc** directory.

If you do not specify a value for STEPLIBLIST, step libraries will not be set up for set-user-ID and set-group-ID executable files.

These step libraries are set up as a result of the invocation of a HFS executable file using the exec service (BPX1EXC), the attach_exec service (BPX1ATX) or spawn (BPX1SPN) service. After one of those services has been invoked, the step libraries can be propagated from the calling task's environment. They can also be specified by using the STEPLIB environment variable that is passed to the exec service. When the exec service invokes a set-user-ID or set-group-ID executable file, only those libraries that are found in the sanctioned list are set up as step libraries in the environment that the executable file will run in.

The following is a list of formatting rules for the STEBLIBLIST file that contains the sanctioned list:

- You can include comment lines in the list. Each comment line must start with /* and end with */.
- You must follow standard MVS data set naming conventions in naming the files in the list.
- Each data set name must be fully qualified and cannot be enclosed in quotation marks.
- Each data set name must be on a line by itself, with no comments.
- You must use uppercase letters for data set names.
- You can put blanks before and after each data set name. Entirely blank lines in the list are ignored.
- You can use the * character to specify multiple files that begin with the same characters. For example, if you list SYS1.*, you are sanctioning any file that begins with SYS1. as a step library.

If the file does not follow these formatting rules, the sanctioned list is not built using the file.

You should catalog each data set listed in the file to prevent user versions of the data set from being used.

Following is a sample sanctioned list file:

```
/********************************************************************/
/*                                                                  */
/*   Name: Sample Sanctioned List for set-user-ID and set-group-ID  */
/*         files                                                    */
/*                                                                  */
/*   Updated by:   May only be updated by OSTEPLIB TSO/E command   */
/*                                                                  */
/*   Description:  Contains a list of data set names that may       */
/*                 be used as STEPLIB libraries for SETUID          */
/*                 programs                                         */
/*                                                                  */
/*                 Wild cards may be used to specify multiple       */
/*                 data set names that have the same prefix         */
/*                 characters.                                      */
/*                                                                  */
/********************************************************************/

/********************************************************************/
/* Sanction all data set names beginning with CEE.SCEERUN           */
/********************************************************************/
CEE.SCEERUN*
```

You can create or update the sanctioned list file using the OSTEPLIB command, which specifies read and execute permissions for all users (permissions 555). The sanctioned list file must be protected from update by nonprivileged users; therefore, only users with superuser authority should be given update access to it.

Because a working copy of the sanctioned list is maintained in storage, an update to the file will take effect when the next setuid(0) program is run from a process with read access to the stepliblist file.

Use the SETOMVS or SET OMVS command to dynamically change the value of STEPLIBLIST; this changes the current system settings. To make a permanent change, edit the BPXPRMxx member that will be used for IPLs.

USERIDALIASTABLE: On most UNIX systems, you use lowercase IDs. With OS/390 UNIX, typically you will use the uppercase user IDs and group names specified in your security database. In some cases, however, you may want to use lowercase or mixed case names in OS/390 UNIX processing. To do that, you need to create a user ID alias table to associate lowercase or mixed case alias names with uppercase OS/390 user ID and group names.

IBM recommends that the pathname of the file be **/etc/tablename**. This fits in with the IBM strategy to place all customized data in the **/etc** directory. If a value for USERIDALIASTABLE is not specified, alias names are not used.

Using the USERIDALIASTABLE statement degrades performance slightly. The more names that you define, the greater the performance degradation. Installations are encouraged to continue using uppercase-only userids and group names defined in their security databases.

Following is a list of formatting rules for the userid alias table:

- You can include comment lines in the list. Each comment line must start with /* and end with */.
- You must follow standard MVS userid and group name naming conventions in the first column.
- You must follow XPG4 standard naming conventions in the second column.
- Do not enclose the names in quotation marks.
- Each userid or group name and associated alias name must be on a line by itself, with no comments.
- The MVS userids and group names must be located in columns 1-8 and the associated aliases must be located on the same line in columns 10-17.
- The MVS name and the alias name must be separated by 1 or more blanks.
- The tags :userids and :groups must be used to delineate between userids and group names.
  - If no tags are present in the file, then all names in the file are assumed to be userids.
  - If there are any names listed before a tag, those names are considered to be userids.
  - If a :userids tag is present, then all name lines following it and up to the next tag are considered to be userids.

- If a :groups tag is present, then all name lines following it and up to the next tag are considered to be group names.
- If specified, the tag must start in column 1.
- The tag names are not case sensitive.

If the file does not follow these formatting rules, the alias name may not be recognized and various functions relating to the attempted use of the alias may fail.

Following is a sample userid and group name alias table:

```
/****************************************************************/
/*                                                               */
/*   Name: Sample user ID/group name alias table                 */
/*                                                               */
/*   Description:  Contains a list of MVS user IDs and their     */
/*                 associated alias names.                       */
/*                                                               */
/* Alias names may be constructed from the following characters: */
/*                                                               */
/*       A B C D E F G H I J K L M N O P Q R S T U V W X Y Z     */
/*       a b c d e f g h i j k l m n o p q r s t u v w x y z     */
/*       0 1 2 3 4 5 6 7 8 9 . _ -                               */
/*                                                               */
/*       The hyphen shall not be used as the first character.    */
/*                                                               */
/*****************************************************************/

/*****************************************************************/
/* Mixed case group names                                        */
/*****************************************************************/
 :Groups
 DEPTD10  DeptD10
 DEPTD20  DeptD20

/*****************************************************************/
/* Non-alphanumeric alias user IDs and group names               */
/*****************************************************************/
 :UserIDs
/*****************************************************************/
/* Mixed case alias names                                        */
/*****************************************************************/
 MYUSERID MyUserid

/*****************************************************************/
/* Easier to remember alias names                                */
/*****************************************************************/
 K61XDLBC Daniel

 JOEDOE   Joe_Doe
 MRDOE    Mr.Doe
 ABCD     A-B-C-D
 :groups
 DEVEL    OE-Dev
 TEST     OE_Test
```

For UUCP, you must set up userid UUCP or define uucp as an alias. Likewise, you must set up group ID UUCPG or define uucpg as an alias.

For more information, refer to . and ..

The userid/group name alias table must be protected from update by non-privileged users; therefore, only users with superuser authority should be given update access to it. All users should be given read access to the file.

Once a user is logged into the system, changing the userid/group name alias table does not change the alias name immediately. Database queries, however, will yield the new alias if the userid performing the query has read/execute access to the userid/group name alias table. The table is checked every 15 minutes and refreshed if it has been changed. If a change needs to be activated sooner, you can use the SETOMVS or SET OMVS command. See 1.1.4.4, “Dynamically Changing the BPXPRMxx Parameter Values” on page 20 for more information.

1.1.4 Chapter 23. Managing Operations

OS/390 UNIX is designed to be continually available. This chapter discusses these tasks, which are done by operators.

For information about the CANCEL, DISPLAY, MODIFY MSGRT, and TRACE operator commands, see *OS/390 MVS System Commands*.

1.1.4.1 Stopping Processes

There are three ways to stop a process:

- The operator enters a MODIFY operator command to terminate a process.
- A shell user enters the **kill** command to cancel processes.
- The operator enters a CANCEL command to stop an address space containing a process. If the address space contains multiple processes, CANCEL terminates all of the processes.

1.1.4.1.1 Terminating a Process with the MODIFY Command: If a process is hung, the operator can enter one of these two MODIFY console commands to terminate the process:

- To allow the signal interface routine to receive control before the process is terminated, issue:

  ```
  F BPXOINIT,TERM=pppp
  ```

  where pppp is the process identifier.

- Sometimes a process is not terminated when a TERM request is sent. In these cases, issue:

  ```
  F BPXOINIT,FORCE=pppp
  ```

  where pppp is the process identifier.

1.1.4.1.2 Terminating a Process with the kill Command: The best way to end a process is to issue the **kill** command. Using the DISPLAY OMVS operator command or the **ps** command, display all the active processes. Then issue the **kill** command, specifying the signal and the PID (process identifier) for the process.

Start by sending a SIGTERM signal:

```
kill -s TERM pid
```

where `pid` is the process identifier. If that does not work, try sending a SIGKILL signal:

```
kill -s KILL pid
```

where `pid` is the process identifier.

1.1.4.1.3 Terminating a Process with the CANCEL Command: An operator can cancel all processes or selected processes in an address space. To cancel all processes, use the CANCEL command. Before issuing CANCEL, display all processes running in that address space and the address space identifier by issuing:

```
DISPLAY OMVS,A=xxxx
```

If there is only one process in the address space or if you want to terminate all the processes, issue:

```
CANCEL name,A=asid
```

For example, for a user with a TSO/E userid of JOE, Figure 2 shows how to obtain the ASIDs for the user's work and then cancel the user's process that is running the **sleep 6000** shell command.

```
display omvs,u=joe
BPXO001I 17.12.23 DISPLAY OMVS 361

OMVS      ACTIVE             OMVS=(93)
USER     JOBNAME  ASID        PID       PPID  STATE   START    CT_SECS
JOE      JOE      001D          5          1  1RI   17.00.10     1.203
JOE      JOE3     001B     131076     262147  1SI   17.00.10      .111
  LATCHWAITPID=          0 CMD=sleep 6000
JOE      JOE1     0041     262147          5  1WI   17.00.10      .595
  LATCHWAITPID=          0 CMD=-sh

cancel joe3,a=1b
```

Figure 2. Console Display for a CANCEL Command

If you want to terminate one or more selected processes in an address space, but not all the processes, then use the MODIFY command as described in 1.1.4.1.1, “Terminating a Process with the MODIFY Command” on page 15 or the **kill** command as described in 1.1.4.1.2, “Terminating a Process with the kill Command” on page 15.

1.1.4.2 Terminating Threads with the MODIFY Command

An operator can terminate a thread, without disrupting the entire process. The syntax of the MODIFY command to terminate a thread is:

```
F BPXOINIT,{TERM}=pid[.tid]
           {FORCE}
```

where

- `pid` indicates the process identifier (PID) of the thread to be terminated. The PID is specified in decimal form as displayed by the D OMVS command.
- `tid` indicates the thread identifier (TID) of the thread to be terminated. The TID is 16 hexadecimal (0-9,A-F) characters as displayed by the following command:

```
D OMVS,PID=pppppppp
```

- TERM= indicates the signal interface routine will be allowed to receive control before the thread is terminated.
- FORCE= indicates the signal interface routine will not be allowed to receive control before the thread is terminated.

  Although abnormal termination of a thread usually causes a process to terminate, using the MODIFY command to terminate a thread will not cause the process to terminate.

You will typically want to terminate a single thread when the thread represents a single user in a server address space. Otherwise, random termination of threads can cause some processes to hang or fail.

If a thread in a process is hung, the operator can enter one of these two MODIFY console commands to terminate the thread without terminating the entire process. We recommend that you use the TERM keyword first, and if that does not succeed, use FORCE:

- To allow the signal interface routine to receive control before the thread is terminated, use:

  ```
  F BPXOINIT,TERM,PID=pppppppp.tttttttttttttttt
  ```

  where pppppppp is the process identifier and tttttttttttttttt is the thread identifier.

- To terminate the thread without allowing the signal interface routine to receive control, use:

  ```
  F BPXOINIT,FORCE,PID=pppppppp.tttttttttttttttt
  ```

  where pppppppp is the process identifier and tttttttttttttttt is the thread identifier.

1.1.4.3 Shutting Down OS/390 UNIX

This section explains how to shut down OS/390 UNIX. When you are doing a planned shutdown and will be re-IPLing the system, issue the following operator command:

```
F BPXOINIT,SHUTDOWN=FORKINIT
```

1.1.4.3.1, “Planned Shutdowns” describes the procedure. If you want to shut down the system as part of JES2 maintenance and do not want to re-IPL the system, use the following operator command:

```
F BPXOINIT,SHUTDOWN=FORKS
```

1.1.4.3.2, “Partial Shutdowns (for JES2 Maintenance)” on page 19 describes the procedure.

1.1.4.3.1 Planned Shutdowns: As part of a planned shutdown, you should clean up the system first before re-IPLing.

1. Use the operator SEND command to send a note to all TSO/E users telling them that the system will be shut down at a certain time. For example:

   ```
   send 'The system is being shut down in five minutes. Log off.',NOW
   ```

2. Use the **wall** command to send a similar note about the impending shutdown to all logged-on shell users. For example:

```
wall The system is being shut down in five minutes. Please log off.
```

3. Prevent new TSO/E logons and shut down other OS/390 subsystems (such as CICS and IMS), following your usual procedures.
4. Shut down all JES initiators.
5. Unmount all NFS-mounted file systems as part of the normal shutdown process.
6. Use normal shutdown procedures to terminate all file system address spaces such as TCP/IP and DFSS. Do this after the final warning has been sent to users that the system is terminating.
7. Terminate running daemons such as **inetd**. To get a list of daemons that are running, issue, for example:

   ```
   D OMVS,U=OMVSKERN
   ```

   In this example, `OMVSKERN` is the userid that is used for the kernel and daemons. In addition, you can display all processes (most daemons will have recognizable names) by issuing:

   ```
   D OMVS,A=ALL
   ```

   Then use the `F BPXOINIT,PID=xxxxxxxx` operator command or the **kill** command to terminate those processes.
8. Terminate any remaining processes and unmount all file systems (including the root file system) by using the bpxstop tool. It is available from the tools and toys page on the OS/390 UNIX web site.

   ```
   http://www.ibm.com/s390/unix/
   ```

9. Take down JES. At this point, there may still be a number of initiators that are provided by WLM for use on fork and spawn. These initiators time out after 30 minutes on their own. To terminate the initiators, you can issue the following operator command:

   ```
   F BPXOINIT,SHUTDOWN=FORKINIT
   ```

10. After all the processes have been terminated, you can do any of the following:
    - IPL
    - Power off
    - Take down JES, restart JES, and then rebuild your environment. For example:
      - Remount any file systems that you unmounted. To do all the mounts, you must issue mount commands or construct a REXX exec or CLIST. If you are using automount for user file systems, there will be less work involved.
      - If you terminated the address spaces for TCP/IP and DFSS, you must restart these.
      - If you terminated daemons, logon to TSO as superuser and run **/etc/rc** from a shell or from the ISHELL.
      - Notify users that the system is once again available for UNIX processing.

1.1.4.3.2 Partial Shutdowns (for JES2 Maintenance): Before JES2 can be shut down for maintenance purposes, part of OS/390 UNIX must be shut down. This section explains how you can terminate all of the forked processes without having to re-IPL the entire system. (The kernel remains active but new forked processes are not allowed.) Use this procedure for JES2 maintenance only.

Do the partial shutdown as infrequently as possible because it is a disruptive shutdown; all the user processes that are either forked or non-local spawned are terminated.

After the forked processes have been terminated, you can terminate the colony address space. Now JES2 can be shut down for maintenance. OS/390 UNIX can be reinitialized after JES2 has been restarted, and forked processes will start being dubbed again. The file system colonies can then be restarted manually. The following steps describe the procedure:

1. Use the operator SEND command to send a note to all TSO/E users telling them that the system will be shut down. For example:

   ```
   send 'The system is being shut down in five minutes. Please log off.'
   ```

2. Use the **wall** command to send a similar note to all logged-on shell users:

   ```
   wall The system is being shut down in five minutes.  Please log off.
   ```

3. Issue the following operator command to begin the shutdown of OS/390 UNIX.

   ```
   F BPXOINIT,SHUTDOWN=FORKS
   ```

   This terminates all forked and non-local spawned address spaces on the system. If the operator receives a success message, the shutdown can be continued.

   A failure message means that some forked processes or non-local spawned address spaces could not be terminated. Try to find these processes by issuing:

   ```
   D OMVS,A=ALL
   ```

   To terminate them, issue:

   ```
   F BPXOINIT,FORCE,PID=xxxxxxxx
   ```

   If that does not work, use the CANCEL or FORCE operator commands.

4. Terminate the file system colonies. Use normal shutdown procedures to close all file system address spaces such as Network File System Client (NFSC) and the Distributed File System Cache Manager (DFSCM).

   For NFSC, determine what the process name was used to start this colony. Use this name to cancel it. (For example, `C NFSC`.)

   For DFSCM, use the procedure in *OS/390 Distributed File Service DFS Administration Guide and Reference* to stop the DFS Cache Manager. Issue `STOP DFSCM` to stop DFSCM.

   For all other colonies, use the procedures documented in their publications.

5. Now you can do whatever corrective or maintenance actions that were needed for JES2, such as restarting it.

6. To restart OS/390 UNIX, issue the Modify (F) command.

   ```
   F BPXOINIT,RESTART=FORKS
   ```

7. Restart the file system address spaces.

   For NFSC, you have to respond to the operator message BPXF014D issued when the colony was taken down. Then reissue all the mounts.

   For DFSCM, respond to the operator message BPXF014D.

   For all other colonies, use the procedures they have documented in their product publications.

1.1.4.4 Dynamically Changing the BPXPRMxx Parameter Values

The SETOMVS command enables you to modify BPXPRMxx parmlib settings without re-IPLing. For example:

```
SETOMVS MAXTHREADTASKS=100,MAXPROCUSER=8
```

You can dynamically change process-wide limits separately for each process. For example:

```
SETOMVS PID=123,MAXFILEPROC=200
```

The SET OMVS command enables you to dynamically change the BPXPRMxx parmlib members that are in effect. Because you can have multiple BPXPRMxx definitions, you can easily reconfigure a large set of the system characteristics. You can keep the reconfiguration settings in a permanent location for later reference or reuse. A sample SET OMVS command is:

```
SET OMVS=(AA,BB)
```

If a parameter is specified more than once with different values, in the parmlib members, the first value specified is the first value that is used. For example, if you specify SET OMVS=(AA,BB) where AA has a MAXPROCUSER=10 value and BB has a MAXPROCUSER=5 value, MAXPROCUSER =10 is used.

You can use the SETOMVS RESET command to dynamically add the FILESYSTYPE, NETWORK, and SUBFILESYSTYPE statements without having to re-IPL. However, if you change the values, a re-IPL will be necessary. For more information, see 1.1.4.4.3, “Dynamically Adding FILESYSTYPE Statements in BPXPRMxx” on page 22.

See *OS/390 MVS System Commands* for a complete description of the SET OMVS and SETOMVS commands.

You can use the SETOMVS SYNTAXCHECK operator command to check the syntax of a BPXPRMxx parmlib member before doing an IPL. (You cannot use that command to verify whether HFS datasets or mount points are valid.)

|	***1.1.4.4.1 Dynamically Changing Certain BPXPRMxx Parameter Values:*** The MAXPROCSYS, MAXPTYS, MAXRTYS, MAXFILEPROC, IPCMSGNIDS, IPCSEMNIDS, IPCSHMNIDS, and IPCSHMSPAGES specify maximum values. You can use the SETOMVS or SET OMVS command to dynamically increase the current system setting, but if you specify a value that is too low or too high, you will get an error message. To use a value outside the range, you will need to change the specification in BPXPRMxx and re-IPL.

To avoid specifying a value that is too low or too high, you can use a formula to calculate the maximum values. The minimum value is sometimes the current setting of the parameter and sometimes lower than that, as identified in the description of each parameter. The formula for each parameter is described later in this section.

The following example shows you how to perform the calculations using the IPCMSGNIDS parameter, which determines the highest number of unique message queues in the system. To use SETOMVS IPCMSGNIDS=*xxx* to increase the current setting, you must calculate the highest number that you can specify. According to the description of IPCMSGNIDS in "IPCMSGNIDS and IPCSEMNIDS" on page 22, the formula is:

```
MIN(20000,MAX(4096,3*initial value))
```

For this example, the current value of IPCMSGNIDS is 1000; the value of IPCMSGNIDS at IPL is also 1000 (that is, 1000 is the initial value). Use the formula in the following way:

1. Compare 4096 with 3 times 1000 to find the higher number (the MAX). 4096 is the higher number.
2. Compare 20000 with 4096 to find the smaller number (the MIN). 4096 is the smaller number.

Therefore, the highest number that you can specify on SETOMVS IPCMSGNIDS is 4096. The range of numbers that you can specify is 1000 (the current value) to 4096. The correct SETOMVS command for increasing the message queue limit to the maximum (assuming a starting value of 1000) would be:

```
SETOMVS IPCMSGNIDS=4096
```

To change to a number higher than 4096 (but lower than 20000), you will have to change BPXPRMxx and re-IPL.

MAXPROCSYS: The range that you can use has a minimum value of 5; the maximum value is based on the following formula:

```
MIN(32767,MAX(4096,3*initial value)
```

The initial value is the MAXPROCSYS value that was specified during BPXPRMxx initialization. You cannot use a value less than 5. If you want to use a value greater than the current maximum (as calculated by the formula) but lower than the initial maximum (32767), you will have to change the value in BPXPRMxx and re-IPL.

MAXPTYS: The range's minimum value is 1 and the maximum is based on the following formula:

```
MIN(10000,MAX(256,2*initial value)
```

The initial value is the MAXPTYS value that was specified during BPXPRMxx initialization.

MAXRTYS: The range's minimum value is the current setting of MAXRTYS, and the maximum is based on the following formula:

```
MIN(10000,MAX(256,2*initial value)
```

The initial value is the MAXRTYS value that was specified during BPXPRMxx initialization. If you want to use a value greater than the current maximum (as calculated by the formula) but lower than the initial maximum (10000), you will have to change the value in BPXPRMxx and re-IPL.

IPCMSGNIDS and IPCSEMNIDS: The range's minimum value is the current setting of IPCMSGNIDS or IPCSEMNIDS, and the maximum is based on the following formula:

```
MIN(20000,MAX(4096,3*initial value)
```

The initial value is the value that was specified during BPXPRMxx initialization. If you want to use a value greater than the current maximum (as calculated by the formula) but lower than the initial maximum (20000), you will have to change the value in BPXPRMxx and re-IPL.

IPCSHMNIDS and IPCSHMSPAGES: The range's minimum value is the current setting of IPCMSGNIDS or IPCSHMSPAGES, and the maximum is based on the following formula:

```
MIN(20000,MAX(4096,3*initial value)
```

The initial value is the value that was specified during BPXPRMxx initialization. If you want to use a value greater than the current maximum (as calculated by the formula) but lower than the initial maximum (20000), you will have to change the value in BPXPRMxx and re-IPL.

1.1.4.4.2 Dynamically Switching to Different BPXPRMxx Members: Another way to dynamically reconfigure parameters is to use the SET OMVS command to change the BPXPRMxx parmlib members that are in effect. With the SET OMVS command, you can have multiple BPXPRMxx definitions and use them to easily reconfigure a set of the OS/390 UNIX system characteristics. You can keep the reconfiguration settings in a permanent location for later reference or reuse.

For example, you could keep the system limits parameters that can be reconfigured in parmlib member BPXPRMLI. When you need to change any of the limits, edit the parmlib member and then issue SET OMVS. For example:

```
SET OMVS=(LI)
```

Changes to system limits (for example, MAXPROCSYS) take effect immediately. Changes to user limits (for example, MAXTHREADS) are set when a new user enters the system (for example, **rlogin** or a batch job). These limits persist for the length of the user connection to OS/390 UNIX.

1.1.4.4.3 Dynamically Adding FILESYSTYPE Statements in BPXPRMxx: Use the SETOMVS RESET command to dynamically add the FILESYSTYPE, NETWORK, and SUBFILESYSTYPE statements without having to re-IPL. If you want to change the values, you will have to edit the BPXPMRxx member that is used for IPLs. You can also dynamically add the parmlib statements currently supported by SETOMVS, such as MAXPROCSYS.

To display information about the current FILESYSTYPE, NETWORK, or SUBFILESYSTYPE statements, issue the following command:

```
DISPLAY OMVS,PFS
```

The following section shows examples of some of the more common configuration changes, adding the HFS and adding sockets. The examples discuss:

1. Activating the HFS file system for the first time.
2. Activating a single sockets file system for the first time.
3. Activating multiple sockets file systems for the first time with Common INET.

4. Adding another sockets file system to an existing common INET configuration.
5. Changing the MAXSOCKETS value.

Activating the HFS File System for the First Time: To activate the HFS file system for the first time, do the following:

1. Set up a root HFS dataset.
2. Create a temporary BPXPRM*tt* member that has the following statement:

   ```
   FILESYSTYPE TYPE(HFS) ENTRYPOINT(GFUAINIT)
   ```

3. Issue `SETOMVS RESET(tt)`.
4. From TSO or the ISHELL, do the following:
   a. Unmount the current root file system.
   b. Mount the root HFS dataset as the new root file system.
   c. Mount any additional HFS datasets as needed.
5. Add the following statements to the BPXPRMxx parmlib member used on IPL:
   a. The FILESYSTYPE statement used above.
   b. A ROOT statement for the root HFS.
   c. MOUNT statements for the additional mounts that should be done initially.

Activating a Single Sockets File System for the First Time: This example explains how to activate a single sockets file system for the first time. It uses the SecureWay TCP/IP Socket File System for network sockets and also brings up support for local sockets. The MAXSOCKETS value used is just an example; the value that you use may be different.

1. Create a temporary BPXPRM*tt* member with the following statements:

   ```
   /* Start Address Family AF_INET for Network Sockets /*
   FILESYSTYPE TYPE(INET) ENTRYPOINT(EZBPFINI)
   NETWORK TYPE(INET)    MAXSOCKETS(2000)
      DOMAINNAME(AF_INET)  DOMAINNUMBER(2)

   /* Start Address Family AF_UNIX for Local Sockets  */
   FILESYSTYPE TYPE(UDS)  ENTRYPOINT(BPXTUINT)
   NETWORK TYPE(UDS)      MAXSOCKETS(1000)
     DOMAINNAME(AF_UNIX)  DOMAINNUMBER(1)
   ```

2. Issue `SETOMVS RESET(tt)`.
3. Start the TCPIP address space.
4. Add these parmlib statements to the BPXPRMxx member used on IPL.

Activating Multiple Sockets File Systems for the First Time with Common INET: This example shows how to activate multiple sockets file systems for the first time with Common INET. It starts two socket file systems, TCP/IP and AnyNet. Because they both support address family AF_INET, they are configured underneath Common INET to give applications the appearance of a single AF_INET socket file system.

Because this is an example of the initial configuration of sockets, the support for local, or AF_UNIX, sockets is also included for completeness.

1. Create a temporary BPXPRM*tt* member with the following statements:

   ```
   /* Start Address Family AF_INET for Common INET */
   FILESYSTYPE TYPE(CINET)  ENTRYPOINT(BPXTCINT)
   NETWORK TYPE(CINET)    MAXSOCKETS(1000)
        DOMAINNAME(AF_INET)   DOMAINNUMBER(2)
        INADDRANYPORT(5000)  INADDRANYCOUNT(100)
   /* Start TCP/IP and AnyNet under Common INET */
   SUBFILESYSTYPE TYPE(CINET)  NAME(TCPIP)  ENTRYOINT(EZBPFINI) DEFAULT
   SUBFILESYSTYPE TIME(CINET)  NAME(ANYNET) ENTRYPOINT(ISTOEPIT)
   ```

2. Issue `SETOMVS RESET(tt)`.
3. Start the TCPIP address space.
4. Start the Sockets Over SNA address space.
5. Add these parmlib statements to the BPXPRMxx member used on IPL.

The names used in the example, `TCPIP` and `ANYNET` must match those used when configuring the associated products.

Increasing the MAXSOCKETS Value: This example shuts down TCP/IP and brings it back up with a new value for MAXSOCKETS:

1. Shut down TCP/IP. For example:

   ```
   p tcpip
   ```

   Most socket programs and daemons will either terminate after TCP/IP is shut down or will tolerate a recycle of TCP/IP. There may be others that will have to be stopped manually.

2. Create a temporary BPXPRM*tt* member that has the following statements:

   ```
   NETWORK TYPE(INET) MAXSOCKETS(10000)
      DOMAINNAME(AF_INET)  DOMAINNUMBER(2)
   ```

3. Issue `SETOMVS RESET=(tt)`.
4. Restart TCP/IP. For example:`S TCPIP`.
5. Restart the socket programs and daemons, as necessary.
6. Update the MAXSOCKETS value in the BPXPRMxx member used on IPL.

Only the SecureWay Socket PFS, EZBPFINI, supports picking up a new MAXSOCKETS value when it is recycled.

The MAXSOCKETS value for a Common INET configuration can be changed with a similar procedure:

1. The TYPE() keyword of the NETWORK statement would specify the TYPE name of the Common INET PFS, which was “CINET” in the previous examples.
2. Common INET is not shut down, though, and the change takes effect in each TCP/IP stack when that stack was recycled.
3. INADDRANYPORT and INADDRANYCOUNT cannot be changed.

Adding Another Sockets File System to an Existing Common INET Configuration: This example starts a second SecureWay Sockets File System and uses names based on the previous examples.

1. Create a temporary BPXPRM*tt* member with the following statements:

```
SUBFILESYSTYPE TYPE(CINET) NAME(TCPIP2) ENTRYPOINT(EZBPFINI)
```

2. Issue `SETOMVS RESET(tt)`.
3. Start the TCPIP2 address space.
4. Add this parmlib statement to the BPXPRMxx member used on IPL.

1.1.4.5 Tracing Events in OS/390 UNIX

To provide problem data, events are traced. When the OMVS address space is started, the trace automatically starts. The trace cannot be completely turned off.

Your installation specifies events to be traced in CTnBPXxx parmlib members. Each member should specify one or more events; keep the number of events small because tracing affects system performance. The installation can filter the events by address spaces, user IDs, and level of detail.

The CTnBPXxx member to be used when the OMVS address space is initialized is identified on the CTRACE parameter of the BPXPRMxx parmlib member. You also specify the size of the trace buffers in the CTnBPXxx member used when the system is IPLed. You can change the buffer size while OS/390 UNIX is running. The buffer can be 16KB minimum to 4MB maximum. If you need a different buffer size, change buffer size (BUFSIZE) in a CTnBPXxx member and issue:

```
TRACE CT,ON,COMP=SYSOMVS,PARM=CTnBPXxx
```

An operator starts and stops tracing events in the OS/390 UNIX system with the commands:

```
TRACE CT,ON,COMP=SYSOMVS,PARM=CTnBPXxx
TRACE CT,OFF,COMP=SYSOMVS
```

The operator can resume full tracing, with the previously used CTnBPXxx parmlib member or a different member, with the command:

```
TRACE CT,ON,COMP=SYSOMVS,PARM=CTnBPXxx
```

The PARM operand specifies the parmlib member with the tracing options.

1.1.4.5.1 Tracing DFSMS/MVS Events: You can also trace DFSMS/MVS events for the HFS. For example, to set up a trace, you can enter the following command:

```
TRACE CT,nnnnk,COMP=SYSSMS
R X,OPTIONS=(CALL,RRTN,CB,SUSP,EXITA,COMP=(ALL,NOIMF,NOSSF)),END
```

or:

```
TRACE CT,nnnnk,COMP=SYSSMS
R X,OPTIONS=(ENTRY,EXIT,EXITA,CB,COMP=(PFS,CDM)),END
```

Attention: SMS trace buffers are allocated in every initiator running kernel workloads. They are allocated in DREF ELSQA, which can cause a shortage of real pages.

For information about how to set up and use a trace, and for diagnosis information on interpreting a trace, see *DFSMS/MVS DFSMSdfp Diagnosis Reference*.

1.1.4.5.2 Re-creating Problems for IBM Service: If you are re-creating a problem for IBM service, it is generally a good idea to increase the OMVS CTRACE buffer size to 4MB. To do this, issue:

```
TRACE CT,4M,COMP=SYSOMVS,PARM=CTnBPXxx
```

with the parmlib member specifying the desired options. Alternatively, you could change the parmlib member to specify the desired buffer size. After you capture the dump for the problem, you can reset the trace buffer size to the original setting. Issue:

```
TRACE CT,xxxK,COMP=SYSOMVS
```

where *xxx*K is the size of the desired trace buffer.

1.1.4.6 Displaying the Status of the Kernel

Display information about the kernel or processes as follows:

- The operator enters a DISPLAY OMVS command to display the status of the kernel and processes.
- The operator enters the DISPLAY TRACE,COMP=SYSOMVS command to display the status of the kernel trace.
- A shell user enters the **ps** command or the PS ISHELL command to display the status of the user's processes.
- A superuser enters the **ps** command or the PS ISHELL command to display the status of all processes.

The operator displays the status for kernel services with the command:

```
DISPLAY OMVS
```

The command can be used to show information about a userid, about the parmlib members that are in effect, or about the current values of reconfigurable parmlib member settings.

To display the status of address spaces that the userid JANES is using and the processor resources used by each address space, the operator enters:

```
DISPLAY OMVS,U=JANES
```

For another example, see Figure 2 on page 16.

If the system IPLed with the specification of OMVS=(XX,YY,ZZ), the output for the D OMVS command is:

```
BPXO004I 10.17.23 DISPLAY OMVS 869
OMVS     ACTIVE  000E        OMVS=(XX,YY,ZZ)
```

The keyword OPTIONS lets you display the current configuration of the BPXPRMxx parmlib statements that are reconfigurable via the SET OMVS or SETOMVS command. The updated output from D OMVS,OPTIONS reflects any changes that resulted from a SETOMVS or a SET OMVS= operator command invocation.

In this example, when the PID option is used to obtain the thread identifiers, the output is:

```
D OMVS,PID=117440514

BPXO040I 14.16.58 DISPLAY OMVS 177
OMVS     000E ACTIVE         OMVS=(93)
USER     JOBNAME  ASID        PID       PPID STATE   START    CT_SECS
MEGA     TC1      0021  117440514  117440515 HKI   14.16.14      .170
  LATCHWAITPID=         0 CMD=ACEECACH
 THREAD_ID        TCB@     PRI_JOB  USERNAME   ACC_TIME SC  STATE
 0496146000000000 009E0438                         .050 PTJ  KU
 04961D0800000001 009D5E88                         .002 SLP JSN
 049625B000000002 009D8798                         .003 SLP JSN
 04962E5800000003 009D5090                         .012 SLP JSN
 0496370000000004 009D5228                         .011 SLP JSN
 04963FA800000005 009D5A88                         .010 SLP JSN
 0496485000000006 009D8048                         .011 SLP JSN
 049650F800000007 009D81E0                         .011 SLP JSN
 049659A000000008 009D8378                         .011 SLP JSN
 0496624800000009 009D8510                         .011 SLP JSN
 04966AF00000000A 009D8930                         .030 SLP JSN
```

You can then cancel selected threads, as shown in this example:

```
F BPXOINIT,FORCE=117440514.04962E5800000003
BPXM027I COMMAND ACCEPTED.

F BPXOINIT,TERM=117440514.0496624800000009
BPXM027I COMMAND ACCEPTED.
```

An operator displays status for the rest of the OS/390 system with the commands:

- **DISPLAY TS,LIST**: The number of time-sharing users, including the number of users
- **DISPLAY JOBS,LIST**: The number of active jobs, including the number of address spaces that were forked or that were created in other ways but requested kernel services.
- **DISPLAY A,LIST**: The combined information from the DISPLAY TS,LIST and DISPLAY JOBS,LIST commands.

1.1.4.7 Taking a Dump of the Kernel and User Processes

If you have a loop, hang, or wait condition in a process and need a dump for diagnosis, you need to dump several types of data:

- The kernel address space.
- Any kernel data spaces that may be associated with the problem.
- Any process address spaces that may be associated with the problem.
- Appropriate storage areas containing system control blocks (for example, SQA, CSA, RGN, TRT).

The steps are:

1. Use DISPLAY commands to display information on currently active address spaces and data spaces. (For more details on these DISPLAY commands, see *OS/390 MVS System Commands*.)

2. Allocate a sufficiently large dump data set.
3. Take the dump.
4. Review the dump completion information.

1.1.4.7.1 Displaying the Kernel Address Space: To find the kernel address space and associated data spaces, use `D A,OMVS`. Here is a sample output:

```
       D A,OMVS
IEE115I 12.55.47 94.208 ACTIVITY 503
 JOBS     M/S    TS USERS    SYSAS    INITS   ACTIVE/MAX VTAM
00001    00013    00002      00019    00019    00002/00050
 OMVS     OMVS     OMVS     NSW  SO  A=000E   PER=NO   SMC=000
                                     PGN=001  DMN=001  AFF=NONE
                                     CT=033.466S  ET=03.44.48
                                     WUID=STC06055 USERID=OMVSKE
                                     ADDR SPACE ASTE=0173ECC0
                                     DSPNAME=SYSZBPXU ASTE=00A35
                                     DSPNAME=SYSGFU01 ASTE=007F8
                                     DSPNAME=SYSZBPX3 ASTE=007F8
                                     DSPNAME=SYSIGWB1 ASTE=007F8
                                     DSPNAME=SYSZBPX2 ASTE=00A35
                                     DSPNAME=SYSZBPX1 ASTE=00A35
```

The display output shows the kernel address space identifier (ASID) as `A=`*nnnn* where *nnnn* is the hexadecimal ASID value. In this example, `A=000E`. The display output also shows the data space names associated with the kernel address space. The system uses these data spaces as follows:

- SYSZBPX1 for kernel data (including CTRACE buffers). The CTRACE buffers are automatically included in the dump and need not be explicitly added to a DUMP command or a SLIP trap.
- SYSZBPX2 for file system data
- SYSZBPX3 for pipes
- SYSIGWB1 for byte-range locking
- SYSGFU01 for file system adapter
- SYSZBPXU for AF_UNIX sockets
- SYSZBPXC for common INET sockets
- SYSZBPXL for local AF_INET sockets

Dump other data spaces if there is reason to believe that they contain data that could be useful in analyzing the problem.

1.1.4.7.2 Displaying Process Information: To display the process information for address spaces, use `D OMVS,A=ALL`. Here is a sample output:

```
    D OMVS,A=ALL

USER     JOBNAME  ASID        PID       PPID STATE
OMVSKERN BPXOINIT 002A          1          0 1WI
MVS      TCPIP    002B      65538          1 MR
DCEKERN  DCEKERN  003A     262147          1 HK
DCEKERN  DCEKERN  003A     262148     262147 HK
DCEKERN  DCEKERN  003A      65541     262147 HK
DCEKERN  DCEKERN  003A      65542     262147 HF
DCEKERN  DCEKERN  003A          7     262147 HK
DCEKERN  DCEKERN  003A          8     262147 HK
TS65106  TS65106  0032          9          1 1RI
TS65106  TS65106  0032         10          9 1CI
  LATCHWAITPID=         0 CMD=-sh
```

The display output shows all of the active processes, ASIDs, process identifiers, parent process IDs, and states. Use this to obtain ASIDs of processes you wish to dump.

1.1.4.7.3 Displaying Global Resource Information: To display global resource serialization information to see possible latch contention, use `D GRS,C`.

This display may show latch contention, which could be the cause of the problem. You should dump the address space of the process holding the latch. If the latch is a file system latch, dump the file system data space SYSZBPX2 also.

1.1.4.7.4 Allocating a Sufficiently Large Dump Data Set: Because you are dumping multiple address spaces, multiple data spaces, and multiple storage data areas, you may need a much larger dump data set defined than is normally used for dumping a single address space. You should preallocate a very large SYS1.DUMPnn data set. For more information on SYS1.DUMPnn data, see the DUMPDS command in *OS/390 MVS System Commands*.

SDUMP has a limit on how much storage it allows in a single dump. It is called MAXSPACE. To determine the current value of MAXSPACE, issue the `D D,O` command. The default value is 500 megabytes. To change this value, issue:

```
CD SET,SDUMP,MAXSPACE=nnnnM
```

In a large server environment, you may need to increase MAXSPACE to 2000M (2 gigabytes) or more.

1.1.4.7.5 Taking the Dump: To initiate the dump, enter this command:

```
DUMP COMM=(dname)
```

where *dname* is a descriptive name for this dump. You can specify up to 100 characters for the title of the dump. The system responds and gives you a prompt ID. You reply by specifying the data to be included in the dump. If you specify the operand CONT, the system will prompt you for more input.

In the following examples of replies you can give, *rn* is the REPLY number to the prompt.

The data areas in the following reply contain system control blocks and data areas generally necessary for investigating problems:

```
R rn,SDATA=(CSA,SQA,RGN,TRT,GRSQ),CONT
```

In the next reply, `x'E'` is the OMVS address space. The other address space IDs specified are those believed to be part of the problem. You can specify up to 15 ASIDs.

```
R rn,ASID=(E,3A,32),CONT
```

This example specifies data spaces:

```
R rn,DSPNAME=('OMVS'.SYSZBPX2,'OMVS'.SYSZBPX1),END
```

The file system data space, SYSZBPX2, is useful if the hang condition appears to be due to a file system latch.

For more information on the DUMP command, particularly on specifying a large number of operands, see *OS/390 MVS System Commands*.

1.1.4.7.6 Reviewing Dump Completion Information: After the dump completes, you receive an IEA911E message indicating whether the dump was complete or partial. If it was partial, check the SDRSN value. If insufficient disk space is the reason, delete the dump, allocate a larger dump data set, and request the dump again.

1.1.4.8 Recovering from a Failure

The operator needs to recover if a failure occurs:

- **Kernel failure:** As a result, interactive processing in the shell and OS/390 UNIX applications fail.
- **File system type failure:** OS/390 UNIX continues processing even though the file system type is not operational. Requests to use the files in any file systems of that file system type will fail.
- **File system failure:** As a result, some files cannot be used, which may cause programs to fail.

The operator starts recovery by collecting messages and a dump, if written.

1.1.4.8.1 System Services Failure: If the OS/390 UNIX system fails, the operator collects problem data, which includes messages, SVC dumps, and SYS1.LOGREC records for abends and decides if re-IPL is warranted.

The work in progress when the failure occurred is lost and must be started from the beginning.

1.1.4.8.2 File System Type Failure: After a failure of a file system type, the system issues message BPXF014D. In response, the operator or automation corrects the problem as indicated by previous messages and then enters `R` in reply to message BPXF014D.

1.1.4.8.3 File System Failure: These events can be symptoms of file system failure:

- 0F4 abend
- EMVSPFSFILE return code
- EMVSPFSPERM return code
- A file becomes unrecognizable or unopenable

After a failure of a file system, the operator:

1. Restores the HFS data set with the data set from the previous level. For more information on recovering an HFS data set, see:
   - *DFSMS/MVS Planning for Installation*
   - *DFSMS/MVS DFSMShsm Storage Administration Guide*
2. Asks a superuser to logically mount the restored HFS data set with a TSO/E MOUNT command.
3. Notifies all shell users that when they invoke the shell they will mount a back-level file system, telling them the mount point. (Use the **wall** command to broadcast a message to all shell users.)

Files added since the back-level data set was saved must be re-created and added again.

If the physical file system owning the root fails, or if the root file system is unmounted, the operator must restore the root file system. This can be done by a superuser who is defined with a home directory of **/**; (root). All work in progress when the failure occurred is lost and must be started from the beginning.

1.1.4.8.4 Recovery of DCE Components: Perform any necessary backup of OS/390 DCE program libraries, configurations, and optional data sets as a part of your regular installation backup and recovery procedures. See *OS/390 DCE Administration Guide* for information about DCE recovery.

1.1.4.9 Managing Interprocess Communication (IPC)

Users can invoke applications that create IPC resources and wait for IPC resources. IPC resources are not automatically released when a process terminates or a user logs off. Therefore, it is possible that an IPC user may need assistance to:

- Remove an IPC resource using the shell's **ipcrm** command
- Remove an IPC resource using the shell's **ipcrm** command to release a user from an IPC wait state

To display IPC resources and which userid owns the resource, issue the following command:

```
ipcs -w
```

To delete message queue IDs, use the **ipcrm -q** or **ipcrm -Q** command.

Another problem may occur when a user waits a long time for a resource such as semaphores or a message receive. Removing a message queue ID or semaphore ID brings any users in an IPC wait state out of the wait state. To display which users are waiting for semaphores and message queues, issue:

```
ipcs -w
```

1.1.5 Chapter 16. Managing the Hierarchical File System

1.1.5.1 Hierarchical File System Concepts

1.1.5.2 Creating a Hierarchical File System

1.1.5.3 Managing File Systems

DFSMS/MVS manages the location of all HFS data sets (file systems) on volumes. However, a file system can outgrow the space on its volume and need more space. Or activity in a file system can become so great that it slows response time. In these cases, the file system needs to be managed.

As of OS/390 Release 7, HFS data sets can span volumes. As users add files and extend existing files, each data set can increase in size to a maximum of 123 extents if secondary extents are specified in the allocation. The system programmer can:

- Remove other data sets from the volume on which the full volume resides.
- Move individual HFS files and subtrees to other volumes.
- Move the entire full file system to another HFS data set.

1.1.5.3.1 If the File System Outgrows Its Space: If the file system becomes too big for the volume, you can try to reduce the size of the file system:

- Create a new file system on another volume and move some files from the full file system to the new file system. Mount the new file system onto the previously full file system.
- Move a subtree from the active file system into a new file system on a different volume. Mount the new file system onto the now-empty directory that was the head of the subtree. Accesses are divided between two volumes.

  Moving a subtree, rather than individual files, retains the hierarchical structure of the file system.

Another approach to making more space available for the file system is to move the entire full file system to another HFS data set, as follows:

1. Have an authorized user enter a TSO/E UNMOUNT command to logically unmount the file system.

   Note: The REXX exec **/usr/sbin/unmount** performs essentially the same functions that the UNMOUNT statement performs. You can run it from the shell.

2. Use the DFSMSdss dump utility to logically dump the old file system to a sequential data set.
3. Rename the old file system.
4. Preallocate a new HFS data set with a larger size and give it the original file system name.
5. Use the DFSMSdss restore utility to restore the old file system to the new data set.
6. Have an authorized user enter a TSO/E MOUNT command to logically mount the new file system. (Or you can run the REXX exec **/usr/sbin/mount** from the shell.)

7. After you have checked the new file system, you can delete the old file system and the corresponding sequential data set.

You can use the **confighfs** command to manage or expand the HFS.

1.1.5.3.2 Removing Unnecessary Files from Directories: You can use the **skulker** OS/390 shell script to remove files that are older than a specified number of days from any directory. It can be run manually or invoked automatically using **cron**.

The **skulker** shell script, which is located in **/samples**, should be copied and can be modified to suit your particular needs. Possible locations for the script include **/bin** or **/usr/sbin**, especially if **skulker** is to be run from a UID 0 program. If **skulker** is to be run by users, **/usr/bin** is another possiblity, but check that the sticky bit is on in the directory.

For more information about **skulker**, see *OS/390 UNIX System Services Command Reference*.

1.1.5.3.3 If the File System Is Too Busy: If activity for a file system becomes so extensive that accesses are slow, do one of the following:

- Move the file system to a volume chosen for speed because it has, for instance, a faster channel or buffered controller.
- Move a subtree from the active file system into a new file system on a different volume. Mount the new file system onto the now-empty directory that was the head of the subtree. Accesses are divided between two volumes.

  Moving a subtree, rather than individual files, retains the structure of the file system.

Chapter 2. APARs OW42811 and OW42841: OS/390 UNIX System Services Command Reference

2.1 skulker— Remove old files from a directory

2.1.1 Format

skulker [**–irw**] [**–l** *logfile*] *directory days_old*

2.1.2 Description

skulker finds files that are candidates for deletion in *directory*, based on the age specified by *days_old*.

When you call **skulker** without any options, the files that are candidates for deletion are found using the primaries as in the following **find** command line:

```
find directory -type f -atime +days_old -level 0 -print
```

For example, specifying 5 for *days_old* causes the **find** command to find files equal to or older than 5 24-hour intervals earlier than now.

The **skulker** script (which is an OS/390 shell script, and can be found in **/samples**) should be copied and can be modified to suit your particular needs. Possible locations for placing the script include **/bin** or **/usr/sbin**, especially if **skulker** is to be run from a UID(0) program. If **skulker** is to be run by users, **/usr/bin** is another possiblity, but check that the sticky bit is on in the directory. If the script is called from a privileged user (a superuser, a user with a UID of 0, or a user running with the RACF **trusted** or **privileged** attribute), it is important to protect the script from any modifications by a non-privileged user.

2.1.3 Options

–i
Displays the files that are candidates for deletion, and prompts the user to stop or continue with file removal. Do not use this option if you are invoking **skulker** from a **cron** job. If **skulker** is invoked with **–i** from a **cron** job, no files will be deleted. A message will be mailed to the caller, showing the **skulker** output that includes the message “Request canceled.”

–l
Specifies a *logfile* to store a list of files that have been deleted, are candidates for deletion, or for which warnings have been mailed; and any errors that may have occurred.

–r
Moves recursively through subdirectories, finding both files and subdirectories that are equal to or older than the specified number of days. The files that are candidates for deletion are found using the primaries as in the following **find** command line:

```
find directory -atime +days_old ! -name directory -print
```

The –name primary prevents **skulker** from deleting the actual directory that was entered as a start point (for example, **/tmp**).

–w Does not remove files, but sends a warning to the owner of each old file (using **mailx**) that the file is a candidate for deletion.

days_old Specifies the age of the files you want to remove. For example, if you specify 100 for *days_old*, all files that were last accessed 100 or more days ago are marked as candidates for deletion.

directory Specifies the directory in which to look for files.

By default, files are removed from the specified directory based on access time and their status as regular files, and are removed only from the directory specified (not from any subdirectories).

2.1.4 Examples

1. To remove all files from **/tmp** that were last accessed 100 or more days ago:

   ```
   skulker /tmp/ 100
   ```

   The trailing slash in **/tmp/** is necessary if **/tmp** is a symbolic link (as it is in OS/390 Release 9 and higher), and you want to list or remove files from the directory the link points to, rather than the symbolic link itself. If **/tmp** (or the directory specified) is not a symbolic link, the trailing slash has no effect.

2. To remove all regular files from **/tmp** that were last accessed 11 or more days ago:

   ```
   > ls -lL /tmp
   total 48
   -rw-------   1 BILLYJC  SHUT           0 Nov 10 06:00 10.txt
   -rw-------   1 BILLYJC  SHUT           0 Nov 11 06:00 11.txt
   -rw-------   1 BILLYJC  SHUT           0 Nov 12 06:00 12.txt
   -rw-------   1 BILLYJC  SHUT           0 Nov 13 06:00 13.txt
   -rw-------   1 BILLYJC  SHUT           0 Nov 14 06:00 14.txt
   -rw-------   1 SUPERID  SHUT           0 Nov 15 06:00 15.txt
   -rw-------   1 BILLYJC  SHUT           0 Nov 16 06:00 16.txt
   -rw-------   1 BILLYJC  SHUT           0 Nov 17 06:00 17.txt
   -rw-------   1 BILLYJC  SHUT           0 Nov 18 06:00 18.txt
   -rw-------   1 BILLYJC  SHUT           0 Nov 19 06:00 19.txt
   > date
   Mon Nov 29 11:17:20 EST 1999
   > skulker -i /tmp/ 11
   -rw-------   1 BILLYJC  SHUT           0 Nov 10 06:00 10.txt
   -rw-------   1 BILLYJC  SHUT           0 Nov 11 06:00 11.txt
   -rw-------   1 BILLYJC  SHUT           0 Nov 12 06:00 12.txt
   -rw-------   1 BILLYJC  SHUT           0 Nov 13 06:00 13.txt
   -rw-------   1 BILLYJC  SHUT           0 Nov 14 06:00 14.txt
   -rw-------   1 SUPERID  SHUT           0 Nov 15 06:00 15.txt
   -rw-------   1 BILLYJC  SHUT           0 Nov 16 06:00 16.txt
   -rw-------   1 BILLYJC  SHUT           0 Nov 17 06:00 17.txt
   -rw-------   1 BILLYJC  SHUT           0 Nov 18 06:00 18.txt
   Do you really want to delete these files?  If yes, answer [y|Y].
   Any other response cancels your request.
   y
   Deleting files...
   > ls -lL /tmp
   total 48
   -rw-------   1 SUPERID  SHUT           0 Nov 15 06:00 15.txt
   -rw-------   1 BILLYJC  SHUT           0 Nov 19 06:00 19.txt
   >
   ```

Note that non-superuser BILLYJC (who issued the **skulker** command) was not able to delete the superuser's (SUPERID) file (`15.txt`), even though the **find** command issued from **skulker** returned `15.txt` as a filename to delete.

3. The **skulker** script can be run from a **cron** job. To use the **cron** daemon to run the **skulker** script at 3:15 a.m. every Monday through Friday:

```
> crontab
15 3 * * 1-5 /etc/skulker -l /usr/spool/cron/skulker.log /tmp/ 100
<control-D>
>
```

This example removes all files from **/tmp** that were last accessed 100 or more days ago. By default, **cron** sends the **stdout** and **stderr** of the command in a mail message to the user who submitted the **cron** job.

2.1.5 Exit Values

0 Successful completion

1 Either **skulker** did not find any files that are candidates for deletion, or an error occurred.

2 There was a usage error.

2.1.6 Messages

Possible messages include:

directory **is not a directory**
The **find** command returned a non-zero exit status: *return code*

Error occurred during remove [of *file***]. Return code=***return code***.**
The rm command failed with *return code* while attempting to delete *file*.

file **is in use, not removed.**
Some other process was using this file. *file* cannot be removed.

Chapter 3. APARs OW42811 and OW42841: OS/390 UNIX System Services User's Guide

3.1 Chapter 15. Working with Files

This chapter covers the topics:

- Using an editor to create a file
- Naming files
- Deleting a file
- Deleting files over a certain age
- Identifying a file by its inode number
- Creating links
- Deleting links
- Renaming or moving a file or directory
- Comparing files
- Sorting file contents
- Counting lines, words, and bytes in a file
- Searching files by using pattern matching
- Browsing files
- Simultaneous access to a file
- Backing up and restoring files
- Listing process IDs of processes with open files

3.1.1 Using an Editor to Create a File

3.1.2 Naming Files

3.1.3 Deleting a File

The command **rm** can delete, or “remove,” several files at once. For example:

```
rm file1 file2 file3
```

removes all the specified files.

Suppose Alice Smith's directory **projectb** had several old meeting notices in it that she wanted to delete: **0607.mtg**, **0615.mtg**, **0623.mtg**, and **0628.mtg**. She could remove all four with just a single command:

```
rm 06*.mtg
```

Be careful when using the wildcard asterisk (*****) for removing files; you may want to use the **-i** option, which prompts you to verify the deletion.

For the tcsh shell, see.for more information on how to control the wildcard asterisk.

3.1.3.1 Deleting files over a certain age

The **skulker** shell script provides a way for the user to delete files based on when the file was last accessed. This can be useful in cases where temporary files created by utilities, or files that were intended to be temporary, but are forgotten about, need to be removed.

The **skulker** script is an OS/390 shell script and can be easily modified to fit any particular system or user needs. The script is located in **/samples**, but the system administrator should have relocated it somewhere else. Users should check with their system administrator for the location of the script. The script should be copied into the user's home directory or subdirectory, where it can be modified by the user if different removal criteria are desired.

It is also possible to invoke the **skulker** script with the **cron** daemon so that it may be run on a regular basis.

The format for running the **skulker** script is as follows:

```
skulker [-irw] [-l logfile] directory days_old
```

directory specifies the directory from which to delete files.

days_old specifies the age of files you want to remove, based on when the file was last accessed.

The **-i** option displays the files to be deleted and then prompts the user to terminate the script or continue with the deletion.

The **-r** option recurses subdirectories, removing both files and subdirectories that are equal to or older than the specified number of days.

The **-w** option does not delete the files, but sends warnings to the owner of each file (via **mailx**) that the file is a candidate for deletion.

The **-l** option allows the user to specify a *logfile* for listing the files that were deleted (or, in the case of the **-w** option, warnings that were sent) and any errors that might have occurred.

For more information on the **skulker** script, see *OS/390 UNIX System Services Command Reference*.

3.1.4 Identifying a File by Its Inode Number

3.1.5 Creating Links

3.1.6 Deleting Links

3.1.7 Renaming or Moving a File or Directory

3.1.8 Comparing Files

3.1.9 Sorting File Contents

3.1.10 Counting Lines, Words, and Bytes in a File

3.1.11 Searching Files by Using Pattern Matching

3.1.12 Browsing Files

3.1.13 Simultaneous Access to a File

3.1.14 Backing Up and Restoring Files: The Options

3.1.15 Listing process IDs of processes with open files

Chapter 4. APARs OW42811 and OW42841: OS/390 UNIX System Services Messages and Codes

FSUM6108 bc: execution error on line *line* **of** *file*

Explanation: Refer to the given line number and file; there may be an error there or a few lines back.

System Action: The program continues.

User Response: See *OS/390 UNIX System Services Command Reference* for more information.

FSUM6630 compress not initialized

Explanation: The program detected that the compression file was not properly initialized.

System Action: The program ends.

User Response: Contact your system programmer.

System Programmer Response: Follow local procedures for reporting a problem to IBM.

FSUM6633 compression not closed

Explanation: The program attempted to open a compression file that was already open and initialized.

System Action: The program ends.

User Response: Contact your system programmer.

System Programmer Response: Follow local procedures for reporting a problem to IBM.

FSUM6635 not initialized

Explanation: The program attempted to close a file that was not initialized.

System Action: The program ends.

User Response: Contact your system programmer.

System Programmer Response: Follow local procedures for reporting a problem to IBM.

FSUM6640 not closed

Explanation: The program attempted to open a file that was already open and initialized.

System Action: The program ends.

User Response: Contact your system programmer.

System Programmer Response: Follow local procedures for reporting a problem to IBM.

FSUM7169 *command***: cannot set access/modify time on** *filename*

Explanation: While *filename* was being restored, the timestamp could not be restored to its original value. This is most likely due to the lack of appropriate permission to the file.

System Action: The program ends.

User Response: Obtain the appropriate permission, or disable the restoring of the timestamp by using the *-pm* option (for pax) or the *-m* option (for tar).

FSUM7172 Warning: file *file***: character** *character*

Explanation: While attempting to convert data in the archive during a read or write, **pax** encountered a character that could not be converted to the desired codeset. The character is left untranslated.

System Action: The program continues.

User Response: See *OS/390 UNIX System Services Command Reference* for more information.

FSUM7207 *command***: "z" (compress) option unavailable with** *option***.**

Explanation: The compress option cannot be specified with the **r** (replace) option; compressed archives cannot be appended to.

System Action: The program ends.

User Response: To add a file to the end of a compressed archive, you can first use the **uncompress** utility to uncompress the archive, and later use the **compress** utility to recompress the archive.

FSUM7221 *command***: file** *hardlink***: hard link to "***targetfile***" ignored: tar format does not permit links to pathnames longer than** *n*

Explanation: The format of the tar and USTAR archives does not allow the storing of files that are hardlinked to files whose name exceeds *n*. Hard links are files that have the same inode value. The first file stored in the archive with the same inode as subsequently archived hardlinks is considered the target of those subsequently archived hardlinks. The name of the target hard link cannot exceed *n* characters, because the format of the archive allows for only *n* characters to represent the name of the target hardlink.

System Action: The program continues.

User Response: For OS/390 Release 9 or later, pax and tar support an extended USTAR format, which can store hard links whose target exceeds *n* characters. See *OS/390 UNIX System Services Command Reference* for more information. For previous versions, the hardlink cannot be stored unless the name of the target hardlink is shortened. For pax, the **-i** or **-s** options can be used to rename files.

FSUM7226 *command***:** *filename***: name too long...switching to USTAR format**

Explanation: The **pax** or **tar** command was using the original UNIX tar format when it encountered *filename*, whose name was too long to be represented in the archive using the original UNIX tar format. It switched to the USTAR format for the remainder of the archive.

System Action: The program continues.

User Response: This is an informational message. No corruption occurred to the archive or to *filename*, and no recovery action is required. To avoid this message, use the **-U** option for tar, to force the use of the USTAR format. For pax, do not use **-x** to specify the tar format; pax will use USTAR as the default.

FSUM7261 *term:* **Unknown terminal type, using dumb.**

Explanation: The terminal type *term* could not be found in the terminal database. The terminal type was set to dumb.

User Response: Check that the **TERM** environment variable is set correctly.

FSUM7333 Use "exit"

Explanation: You entered an end-of-file (EOF) character (Ctrl-D), but the **ignoreeof** shell option is set. The **ignoreeof** option tells the shell not to exit when an EOF character is entered.

System Action: The shell continues.

User Response: Enter the **exit** command to exit the shell.

FSUM7338 execute: internal error (*number*)

Explanation: This is an internal error in the OS/390 shell.

System Action: The command ends.

User Response: See your IBM service representative.

FSUM7350 e_cmd: negative result?

Explanation: This is an internal error in the OS/390 shell.

System Action: The command ends.

User Response: See your IBM service representative.

FSUM7428 expression: internal error

Explanation: This is an internal error in the OS/390 shell.

System Action: The command ends.

User Response: See your IBM service representative.

FSUM7455 *command name*: warning--file size error in *filename* --file truncated

Explanation: While *filename* was being written to the archive, the size of the file grew. Only the amount of the file that corresponds to the original size of *filename* was archived; the contents of *filename* in the archive no longer match the contents of *filename* on your file system.

System Action: The program continues.

User Response: If the truncation of *filename* is a problem, rebuild the archive.

FSUM7456 *command name*: warning--file size error in "*filename*"--file padded with spaces

Explanation: While *filename* was being written to the archive, another unrelated application caused the size of the file to become smaller. *filename* was stored in the archive padded with spaces to its original size, and the contents of *filename* in the archive no longer match the contents of *filename* on your file system.

System Action: The program continues.

User Response: If the padding of *filename* is a problem, rebuild the archive.

FSUM7895 Failed to preserve file.

Explanation: **vi** attempted to write a recoverable version of the current working file to **/etc/recover/$LOGNAME**, using the **exrecover** utility. It is possible that the **exrecover** command failed, or that the **/etc/recover** directory is not configured properly.

System Action: The program continues.

User Response: To be sure you have a backup copy of your file, issue a **:w** *alternatefilename* to write out the current version of the file with this new (not existing) name. Contact your system programmer for further assistance.

System Programmer Response: Verify that the **/etc/recover** directory exists, and that it has proper permissions. Verify that **/tmp** (or wherever **vi** temporary files are currently being written) is not corrupted or full. For further assistance, follow local procedures for reporting a problem to IBM.

FSUM7911 Global within global not allowed.

Explanation: You specified either a **g** or a **v** as the command to be used with the **ex** global command. The global commands **g** or **v** cannot occur in the list of commands to run globally.

System Action: The program continues.

User Response: Verify that the command you entered has no errors, or attempt your action through another method. See *OS/390 UNIX System Services Command Reference* for more information.

FSUM7960 Internal error: ltsave().

Explanation: **vi**'s internal tables, which store line table information, have been corrupted.

System Action: The program continues.

User Response: If you want to try to save the latest changes, use **:w** *newfilename* (where *newfilename* does not already exist) to avoid overwriting the current file. Check the original file and the new file (*newfilename*) for corruption, and contact your system programmer.

System Programmer Response: Follow local procedures for reporting a problem to IBM.

FSUM8211 Null to Expand

Explanation: The system encountered an internal error while trying to expand a null string.

System Action: The program ends.

User Response: Contact your system programmer.

System Programmer Response: Follow local procedures for reporting problems to IBM.

FSUM8233 Illegal parser state *state*

Explanation: Internal error.

System Action: The program ends.

User Response: Contact your system programmer.

System Programmer Response: Follow local procedures for reporting problems to IBM.

FSUM8718 *name***: cannot open** *string***:**

Explanation: Could not open the file. This message is followed by a system error describing the specific reason for the error.

System Action: The program ends.

User Response: Refer to the system error message and take appropriate action.

FSUM8765 Internal error: glob routine: Code *code*

Explanation: While attempting to get a list of all recovered files, the C/C++ Run-Time Library function **glob()** failed, for the reason specified by *code*. It is possible that an attempt to allocate memory failed.

System Action: The program continues.

User Response: The return codes of **glob** are specified in **glob.h**. Look in **glob.h** to find the exact reason for the failure, and attempt the command again.

FSUM8913 unexpected end of file.

Explanation: The end of the file was encountered prematurely.

System Action: The program ends.

User Response: Check the input file for truncation.

FSUM9153 Cannot chdir to *filename*

Explanation: Could not change directory to *directory*.

System Action: The program ends.

User Response: The system error displayed with this message indicates the cause.

FSUM9154 Shell (*filename*) not executable.

Explanation: The **newgrp** utility attempted to invoke a new shell named *filename*, but it could not be run.

System Action: The **newgrp** command ends, and the shell from which **newgrp** was issued is terminated, because **newgrp** has already replaced the shell's process image.

User Response: Check that *filename* is a valid shell (command interpreter) and that it has the correct permissions.

FSUM9169 Internal, buildList buffer too small

Explanation: Internal error.

System Action: The program ends.

User Response: Contact your system programmer.

System Programmer Response: Follow local procedures for reporting problems to IBM.

FSUM9453 *library(member)* : Can't touch library member

Explanation: Warning: The command failed when trying to touch the library member to force the modify time to the present.

System Action: The program continues.

User Response: Check the library and try again.

FSUM9593 dd: only one of conv=ucase and conv=lcase may be specified

Explanation: You specified both **ucase** and **lcase**. These options are mutually exclusive; you can specify only one.

System Action: The program ends.

User Response: Select **ucase** or **lcase** and rerun the command.

FSUM9594 dd: only one of conv=block and conv=unblock may be specified

Explanation: You specified both **block** and **unblock**. These options are mutually exclusive; you can specify only one.

System Action: The program ends.

User Response: Select **block** or **unblock** and rerun the command.

FSUM9595 dd: only one character set translation option may be specified

Explanation: The **ascii**, **ebcdic**, **ibm**, and **convfile** translation options are mutually exclusive. You specified more than one of these options.

System Action: The program ends.

User Response: Select only one translation option and rerun the command.

FSUM9602 unable to create tempfile *name*

Explanation: A temporary file could not be created. A system error indicating the cause follows this message. The problem is usually caused by lack of space on the output device, or lack of appropriate permissions to create the file.

System Action: The program ends.

User Response: Refer to the system error message and take appropriate action.

FSUM9603 output path or file name too long

Explanation: The pathname of the temporary file exceeds system limits.

System Action: The program ends.

User Response: Because the name of the temporary file is based on the pathname of the input file, shorten the pathname of the input file by renaming it and/or moving it to a directory with a shorter name.

FSUM9604 file *filename* **already exists; not overwritten**

Explanation: The output file *filename* already exists and will not be overwritten.

System Action: The program ends.

User Response: Remove or rename *filename*.

FSUM9620 File read error

Explanation: **ed** experienced a problem reading the file. A system error message follows.

System Action: The program continues.

User Response: Respond to the accompanying system error message.

FSUM9636 System does not support querying the set of character sets

Explanation: The **-l** option is not supported.

System Action: The program ends.

User Response: Consult your system programmer if this is a needed feature.

FSUM9637 Warning: multibyte locale not supported

Explanation: **lex** currently does not support multibyte locales. If you are attempting to run **lex** in a multibyte locale, you may experience additional errors. See *OS/390 UNIX System Services Command Reference* for more information.

System Action: The program continues.

User Response: Run **lex** in POSIX locale.

FSUM9638 Number *num* **not in range** *min***.** *max*

Explanation: The number you specified, *num*, is not within the allowable range of numbers.

System Action: The program ends.

User Response: Specify a number in the range *min* and *max*.

FSUM9639 Number *number* **not in range** *0* **...***unsigned long max*

Explanation: Input data fell outside of the accepted range, because you attempted to enter a negative number.

System Action: The program ends.

User Response: Enter only non-negative numbers.

FSUM9661 pipe buffer

Explanation: While attempting to pipe a message through a command, **mailx** was not able to allocate enough memory.

System Action: The program continues.

User Response: Free up system resources and retry the command.

FSUM9666 No messages satisfy : *subcommand line*

Explanation: No messages in the mailbox matched the subcommand given.

System Action: The program continues.

FSUM9667 No applicable messages about *subcommand line*

Explanation: No messages in the mailbox had a subject line that matched the one given by the user.

System Action: The program continues.

FSUM9668 No applicable messages from *subcommand line*

Explanation: The system could not find a mail message from the specified user.

System Action: The program continues.

FSUM9669 Unrecognized scrolling command *command*

Explanation: The user gave a scrolling command that was not valid.

System Action: The program continues.

User Response: To scroll forwards, use **+** or **z+**. To scroll backwards, use **-** or **z-**.

FSUM9673 *pathname***: Nonportable character** *c* **(***xx***) found.**

Explanation: *pathname* contains the character *c* (represented in hex by *xx*), which is not in the portable filename character set.

System Action: The program ends.

User Response: Rename *pathname* so that it does not include this character.

FSUM9674 *pathname***: Nonportable byte** *xx* **found.**

Explanation: *pathname* contains the non-printable character with hex byte value of *xx*, which is not in the portable filename character set.

System Action: The program ends.

User Response: Rename *pathname* so that it does not include this character.

FSUM9679 Existing file *filename* **exists; it will not be overwritten**

Explanation: The **-k** (do not overwrite) option was specified for a **pax** restore, and **pax** has determined that *filename* already exists on the file system. The version of *filename* in the archive will not be restored.

System Action: The program continues.

User Response: This is an informational message. No response is required.

FSUM9681 *n* **illegal character sequence(s) for codeset extracting file** *filename*

Explanation: While *filename* was being extracted, *n* characters could not be translated from and to codesets specified on the **-o to=** option. The *n* characters are left untranslated.

System Action: The program continues.

User Response: Verify that the *filename* as stored in the archive is not corrupted, and use the **iconv** utility to verify that the file can be successfully converted in its original form.

FSUM9682 Cannot append to compressed archive

Explanation: The **-a** (append) option cannot be used on a compressed archive.

System Action: The program ends.

User Response: To add a file to the end of a compressed archive, you can first use the **uncompress** utility to uncompress the archive, and later use the **compress** utility to recompress the archive.

FSUM9696 !write error on file *filename*

Explanation: An error occurred writing to *filename*. A system error message indicating the reason for the error follows this message. The problem is usually caused by lack of space on the output device, or lack of appropriate permissions to write the file.

System Action: The program ends.

User Response: Refer to the accompanying system error message and take appropriate action.

FSUM9797 Badly formed sort key position *position*

Explanation: The key position was not specified correctly.

System Action: The program ends.

User Response: Check the format and try again.

FSUM9698 Must specify number in *-option*

Explanation: Only numeric input data is accepted.

System Action: The program ends.

User Response: Use proper numeric data and try again.

FSUM9700 too many key field positions specified

Explanation: **sort** is limited in the number of key positions that can be specified. See *OS/390 UNIX System Services Command Reference*.

System Action: The program ends.

User Response: Use fewer key fields.

FSUM9701 key value in *string* out of bounds

Explanation: The key value must be between **1** and **LINEMAX**.

System Action: The program ends.

User Response: Change the key value.

FSUM9702 invalid key specification *key*

Explanation: If you specify a key using *m.n* notation, you cannot specify `0` in the *n* position.

System Action: The program ends.

User Response: Respecify the command using proper key syntax. See *OS/390 UNIX System Services Command Reference* for more information.

FSUM9704 file *filename*: no newline at end of file

Explanation: The file did not end in a newline character. A newline character was added.

System Action: The program continues.

User Response: To avoid this message in the future, add a newline to the end of the file.

FSUM9705 file *filename*: line too long: limit *max* -- truncated

Explanation: The line exceeded the allowed length, and as a result was truncated. This could be due to a missing newline character.

System Action: The program continues.

User Response: To avoid this message in the future, make sure the line length does not exceed the stated limit and that the file is not missing newline characters.

FSUM9707 file *filename*: line *linenumber*: non-unique key in record: *record*

Explanation: The field selected is not suitable as a key field because its values are not unique for each record.

System Action: The program ends.

User Response: Select a different key field and try again.

FSUM9708 file *filename*: line *linenumber*: not ordered properly at: *record*

Explanation: The fields of the given record were apparently out of sequence in comparison with previous records.

System Action: The program ends.

User Response: Check for missing field(s), or reorder the given record to match the previous records.

FSUM9710 !temporary file error *filename*

Explanation: **crontab** was unable to open the named temporary file. This could be due to insufficient free storage, or a problem with the **/tmp** directory.

System Action: The program ends.

User Response: Make sure that **/tmp** exists and is not full, try freeing up system resources, and retry the command.

FSUM9712 Upper/lower case conversion must be specified in the same relative positions

Explanation: The upper or lower character class specified in *string2* does not correspond to the opposite lower or upper class specified in *string1*.

System Action: The program ends.

User Response: Change the position of the upper or lower character class in *string2* to correspond with the lower or upper character class in *string1*.

FSUM9922 internal execution tree error at *string*

Explanation: **awk** encountered an error at the named step.

System Action: The program continues in an error state and may end later.

User Response: See *OS/390 UNIX System Services Command Reference* for more information on **awk**.

FSUM9926 error reading file

Explanation: An unspecified error occurred in trying to read the file. A system error message that explains the error follows this message.

System Action: The program continues.

User Response: Respond to the system error message.

FSUM9927 error splitting record: *line*

Explanation: **awk** encountered a syntax error on the given line.

System Action: The program continues.

User Response: Correct the error and retry. For more information, see *OS/390 UNIX System Services Command Reference*.

FSUM9928 invalid wide character *hex character code*

Explanation: **awk** encountered a character that was not a valid wide character.

System Action: The program continues.

User Response: Contact your system programmer.

System Programmer Response: Follow local procedures for reporting a problem to IBM.

FSUM9942 *argument* **ignored, same basename as** *argument***.**

Explanation: Duplicate arguments are not allowed.

System Action: The program continues.

User Response: To avoid this message in the future, remove the duplicate argument.

FSUM9943 file(s) contain a character outside the domain of the collating sequence

Explanation: One or more characters in the file(s) has an incorrect character, based on information in the **LC_COLLATE** category of the current locale.

System Action: The program continues.

User Response: Review the file(s) for incorrect characters (see **LC_COLLATE**).

FSUM9948 Warning: useless variables:

Explanation: The listed variables are either redundant or not used.

System Action: The program continues.

User Response: Review your code to see if the listed variables can be eliminated, or if the code that uses them is missing.

FSUM9949 invalid character *character*

Explanation: **awk** encountered a character that it did not recognize as alphanumeric, whitespace, or a special character to **awk** (such as a metacharacter).

System Action: The program ends.

User Response: Check your input for characters that are not valid, and check that the codeset of the input data matches the codeset of the shell environment.

FSUM9952 history file $HISTFILE "*file***"**

Explanation: The OS/390 shell command history file specified by the **HISTFILE** variable could not be opened.

System Action: The shell continues to run with the default history file: **$HOME/.sh_history**.

User Response: Take appropriate action based on the specific error description at the end of this message.

FSUM9954 bad month *month number*

Explanation: Months range from 1 to 12.

System Action: The program ends.

User Response: Enter again with a valid month number.

FSUM9955 invalid year *year*

Explanation: Years range from 1 to 9999.

System Action: The program ends.

User Response: Enter again with a valid year.

FSUM9956 bad month *month*

Explanation: The month name, spelling, or abbreviation was incorrect.

System Action: The program ends.

User Response: Check the spelling of the month, or use the first 3 letters of the month name as an abbreviation.

FSUM9988 !cannot determine PATH_MAX

Explanation: The program was unable to get the size limit for filenames (including their directory path) from the system.

System Action: The program ends.

User Response: Contact your system programmer.

System Programmer Response: Follow local procedures for reporting problems to IBM.

FSUM9989 invalid queuename *queuename*

Explanation: *queuename* must be a single-byte character long, and must not be a space, tab, newline, or NULL.

FSUM9991 time specified has already expired

Explanation: You specified a time that was in the past.

User Response: Reissue the command with a correct time.

FSUM9992 must run as root, and be run by root

Explanation: **cron** can only be started by a superuser. See *OS/390 UNIX System Services Command Reference* for more detailed information.

System Action: The program ends.

User Response: Contact your system administrator to start **cron**.

FSUM9993 cron already running, pid #*pid number*

Explanation: You may only start **cron** once, and it is already running.

System Action: The **cron** that is already running continues to run.

FSUM9994 cron already running

Explanation: You may only start **cron** once, and it is already running.

System Action: The **cron** that is already running continues to run.

FSUM9995 bad format in queuedefs

Explanation: There is a bad format in the **queuedefs** file. The queue format is *quename.#j#n#w*, where *quename* is a single character and *#* is a numeric value followed by a letter: **j** for the maximum number of jobs, **n** for null value, or **w** for wait to reschedule.

System Action: The program ends.

User Response: Check the **queuedefs** file for errors. See *OS/390 UNIX System Services Command Reference* for more details.

FSUM9996 insufficient permission for -u

Explanation: The **-u** option can only be used by a superuser.

System Action: The program ends.

FSUM9997 You are not authorized to use cron. Sorry.

Explanation: Either your user ID is not in **cron.allow**, or it is in **cron.deny**.

User Response: Contact your system administrator.

FSUM9998 can't remove your crontab file

Explanation: **crontab** was unable to remove the **crontab** file.

System Action: The program ends.

User Response: Check the permissions on the **crontab** file and verify that write permission has been granted.

FSUMA003 Unexpected character found in line: *line*

Explanation: **crontab** was expecting to find either a valid number or an ***** character followed by a non-blank (space or tab) character.

System Action: The program continues.

User Response: Check the input line. See *OS/390 UNIX System Services Command Reference* for the correct format.

FSUMA004 Number out of bounds in line: *line*

Explanation: **crontab** encountered a number in the input line that is outside of the allowed range.

System Action: The program ends.

User Response: Check the input line. See *OS/390 UNIX System Services Command Reference* for the correct format.

FSUMA005 Missing command on line: *line*

Explanation: **crontab** was expecting to find a command on the given input line and failed to detect one.

System Action: The program ends.

User Response: Check the input line for incorrect format.

FSUMA007 Warning: *string* **already defined in** *filename*

Explanation: A duplicate identifier was found in the given file.

System Action: The program continues.

FSUMA008 DLL in use

Explanation: The system is unable to run the program while the DLL is in use by another program or another instance of this program.

System Action: The program ends.

User Response: Make sure **diff** is not running in a background process (use **ps** to detect this). If it is, either use the **kill** command to end it or use the **fg** command to bring it to the foreground. If the problem persists, contact your system administrator.

FSUMA009 *filename***: Not a directory**

Explanation: **dircmp** can only be used to compare directories.

System Action: The program ends.

User Response: If you are comparing files, use **diff** or **cmp**; otherwise, verify that both arguments to **dircmp** are directories.

FSUMA081 database file *file*

Explanation: The program was unable to open the named file for reading.

System Action: The program continues.

User Response: Check to make sure that the file exists and that you have permission to read it.

FSUMA084 Multibyte conversion error.

Explanation: An error occurred in converting between wide-character and multibyte types. It is possible that an incorrect multibyte character was encountered.

User Response: Check the data for incorrect multibyte characters, and try again.

FSUMA093 write error on standard output

Explanation: The program was unable to write to standard output.

System Action: The program ends.

User Response: Contact your system programmer.

System Programmer Response: Follow local procedures for reporting problems to IBM.

FSUMA094 *file* **is not a normal file--can't patch.**

Explanation: You attempted to patch a file that is not a regular file.

System Action: The program ends.

User Response: Make sure there were no typographical errors in the filename, then check the file itself for errors.

FSUMA096 Not enough memory to try swapped hunk! Assuming unswapped. This may be due to insufficient storage space in the system.

Explanation: **patch** was unable to allocate memory from the system.

System Action: The program continues.

User Response: Free up system resources and retry the command.

FSUMA099 Hunk *number* **failed at** *line number*

Explanation: **patch** was unable to process the given hunk. Unless you specify otherwise, those hunks that failed will be saved in a file in the current directory.

System Action: The program continues.

User Response: It is possible that the patchfile was applied to the wrong file. Double-check patch files, source files, and any reject files that were created to ensure proper order.

FSUMA104 Ran out of memory using Plan A--trying again...

Explanation: **patch** was unable to allocate enough memory to perform an action.

System Action: The program continues, attempting to apply the remaining patches (hunks) from the file.

FSUMA107 You may not change to a different patch file.

Explanation: Only one patch file may be used per patch.

System Action: The program continues.

User Response: See *OS/390 UNIX System Services Command Reference* for more information.

FSUMA108 Unable to change directory to *directory*

Explanation: The directory you specified may not exist; or you may have insufficient permissions to access it.

System Action: The program continues.

User Response: Double-check the directory path and its permissions.

FSUMA110 Fatal internal error in abort_hunk()

Explanation: While attempting to output information from a failing hunk, **patch** encountered a character it did not recognize.

System Action: The program ends.

User Response: Check the patch file for corruption or truncation. Also check that the codeset in which the patch file is encoded matches that of the current environment.

FSUMA111 Out-of-sync patch, lines *starting line, ending line* --mangled text or line numbers, maybe?

Explanation: While attempting to apply the current hunk, **patch** encountered an error.

System Action: The program ends.

User Response: Check the patch file for corruption.

FSUMA116 This appears to be the wrong kind of patch.

Explanation: The **diff** type you specified did not match the actual type of the patch.

System Action: The program continues.

User Response: Respecify the **diff** type (**context**, **ed**, or **normal**) so that it matches the actual type of the file. For more information, see *OS/390 UNIX System Services Command Reference* under **diff** and **patch**.

FSUMA130 Unexpected end of file in patch.

Explanation: While processing a segment of the patch file, **patch** ran across an unexpected end of file. The patch file could be corrupted.

System Action: The program ends.

User Response: Recreate the patch file, restore the file to be updated from backup, and rerun the program.

FSUMA131 Unexpected end of hunk at line *line number*.

Explanation: **patch** came across the end of the current hunk (segment) of the patch file sooner than expected.

System Action: The program continues.

User Response: Examine the patch file for errors.

FSUMA132 Unexpected * at line** *line number*: *line*

Explanation: **patch** came across the start or end of a hunk or segment sooner than expected.

System Action: The program continues.

User Response: Examine the patch file for errors.

FSUMA133 Duplicate --- at line *line number* **--check line numbers at line** *line number*.

Explanation: **patch** came across the start or end of a hunk or segment sooner than expected. This may be due to an extraneous **---** marker.

System Action: The program continues.

User Response: Examine the patch file for errors.

FSUMA134 Premature --- at line *line number*--**check line numbers at line** *line number*.

Explanation: The **---** indicator of the next section of the patch file was encountered sooner than expected.

System Action: The program continues.

User Response: Check the line numbers specified; the patch file may have been edited or otherwise corrupted.

FSUMA135 Overdue --- at line *line number*--**check line numbers at line** *line number*.

Explanation: The **---** indicator of the next section of the patch file was expected and not found.

System Action: The program continues.

User Response: Check the line numbers specified; the patch file may have been edited or otherwise corrupted.

FSUMA136 Hunk too large (*quantity* **lines) at line** *line number*: *line*

Explanation: The current hunk exceeds the maximum allowable hunk size.

System Action: The program continues.

User Response: Consult your system programmer.

System Programmer Response: Follow local procedures for reporting problems to IBM.

FSUMA137 No --- found in patch at line *line number*

Explanation: The **---** marker, which marks the beginning of replacement text, was expected and not found.

System Action: The program continues.

User Response: Examine the patch file for corruption.

FSUMA139 Replacement text or line numbers mangled in hunk at line *line number*

Explanation: The patch file may have been corrupted or applied in the reverse order.

System Action: The program continues.

User Response: Examine the patch file for corruption.

FSUMA140 Unexpected end of file in patch at line *line number*

Explanation: **patch** encountered the end-of-file marker before it had finished processing the patch file.

System Action: The program continues.

User Response: Check the patch file for corruption or truncation.

FSUMA144 Not enough memory to swap next hunk!

Explanation: This message occurs only when you use the **-R** option on the command line. **patch** was unable to allocate memory to swap a hunk to or from the patch file.

System Action: The program continues.

User Response: To avoid this problem in the future, try freeing up system resources.

FSUMA145 Malformed patch at line *line number*: *line*

Explanation: **patch** was expecting to find a digit and did not find one.

System Action: The program ends.

User Response: Check the patch file for errors.

FSUMA157 at .so *filename*, file nesting level too deep

Explanation: You attempted to push more than 64 files onto the stack.

System Action: The program ends.

User Response: Try breaking up the **spell** command so that you do not need to examine more than 64 files at one time.

FSUMA162 hash table

Explanation: The system was unable to allocate storage for the hash table.

System Action: The program ends.

User Response: Free up system resources, or add more main storage.

FSUMA163 hash file *file*

Explanation: The program was unable to open the named file.

System Action: The program ends.

User Response: Check the file permissions. If the failure persists, contact your system administrator.

FSUMA164 hash file *file* write

Explanation: The program was unable to write to the named file.

System Action: The program ends.

User Response: Refer to the system error message that accompanies this message.

FSUMA176 Incomplete multibyte sequence on command line

Explanation: The multibyte character sequence specified in the input or output translation string is incomplete.

System Action: The program ends.

User Response: This is most likely due to a missing **\SI** (shift in) character following a **\SO** (shift out).

FSUMA186 Internal error: wcfputs().

Explanation: In attempting to write a wide-character string to a stream, an internal error occurred during conversion to multibyte.

System Action: The program continues.

User Response: Consult your system programmer, or follow local procedures for reporting a problem to IBM.

System Programmer Response: Follow local procedures for reporting a problem to IBM.

FSUMA187 Failed to expand words.

Explanation: **vi** was unable to expand the **%**, **#**, **!** commands on the input line.

System Action: The program continues.

User Response: Check the input line for mistyped arguments or files.

System Programmer Response: None.

FSUMA194 Problem creating pathname (*path*)

Explanation: This is a system error that may have been caused by either of the following situations (although other causes are possible):

1. The pathname length exceeds **PATH_MAX**.
2. There was insufficient memory to create the pathname.

System Action: The program ends.

User Response: Contact your system programmer.

System Programmer Response: Follow local guidelines for reporting the problem to IBM.

FSUMA196 temporary file error

Explanation: **wall** was unable to open a temporary file. There could be insufficient system resources; for example, not enough storage or open file pointers.

System Action: The program ends.

User Response: Free up system resources and try again if necessary.

FSUMA197 write error on temporary file

Explanation: An error occurred while the program was attempting to write to a temporary file.

System Action: The program ends.

User Response: Make sure that system resources have not been used up, as insufficient free storage could cause this problem.

FSUMA860 Recursive alias *alias* not expanded

Explanation: This alias was part of another alias, and exceeded the maximum recursive depth allowed.

System Action: The program continues.

User Response: Redefine the alias so that an alias name is not part of the definition.

FSUMA873 !cannot create or open file for writing *pathname*

Explanation: *pathname* could not be opened. A system error indicating the cause follows this message. The problem is usually caused by lack of space on the output device, or lack of appropriate permissions to create the file.

System Action: The program ends.

User Response: Refer to the system error message and take appropriate action.

FSUMA876 invalid number (*num***) specified with -n option (max** *max* **allowed)**

Explanation: The number *num* that was specified for the **-n** option exceeds the allowable value *max*.

System Action: The program ends.

User Response: Respecify a number that is less than or equal to *max*.

FSUMA877 !read error on file *pathname* **at line** *n*

Explanation: An error occurred in file *pathname* at line *n* that prevented the file from being read. A system error indicating the cause follows this message.

System Action: The program ends.

User Response: Refer to the system error message and take appropriate action.

FSUMA878 file *pathname***, line** *n***: contains binary data**

Explanation: *pathname* at line *n* contains data that is not text. This program requires text data.

System Action: The program ends.

User Response: Verify that *pathname* is a text file.

FSUMA879 file *pathname***, line** *n***: line length exceeds limit of** *max*

Explanation: The length of line *n* in *pathname* exceeds the maximum allowed value *max*.

System Action: The program ends.

User Response: Because the name of *pathname* is based on the pathname of the input file, shorten the pathname of the input file by renaming it and/or moving it to a directory with a shorter name.

FSUMA882 *string* **: Premature end of file**

Explanation: **csplit** encountered the end of file before it was expected. This usually indicates that the last line of the input file is not terminated by a line-end character.

System Action: The program ends.

User Response: **csplit** operates on text files. Verify that the file consists of complete text lines.

FSUMA883 *string* **: Out of range**

Explanation: This is most likely the result of specifying a negative offset from a regular expression, which resulted in a negative number of lines to output.

System Action: The program ends.

User Response: Reattempt the command with a smaller negative offset, or change the regular expression.

FSUMA884 !memory allocation error

Explanation: You ran out of memory.

System Action: The program ends.

User Response: Obtain more memory.

FSUMA885 find: could not execute cpio utility

Explanation: **popen** failed to execute **cpio**. Another message describing the error is displayed.

System Action: The program ends.

User Response: Look up the associated error message.

FSUMA891 No such user as *user*

Explanation: The user specified with the **-u** option is not known on this system. Either the user name or the numeric user ID is not defined in the user database.

System Action: The command ends.

User Response: Specify a valid user name or user ID.

FSUMA897 pathchk: warning: pathconf(*pathname* **,***varcode***) returns** *syserror***. Using** *varcode2***=** *rc*

Explanation: An error occurred executing the C runtime function **pathconf()** on *pathname* using varcode=*varcode*.

System Action: The program ends.

User Response: Refer to the system error message and take appropriate action. Contact your system programmer for additional help.

FSUMA911 undefined function

Explanation: The command is identified as an undefined function. Possible causes:

1. The command name was marked with the **typeset -u** or **autoload** command, and the definition was not found in **FPATH**.
2. A filename matching the command name was found in **FPATH**, but the file did not contain a function definition of that name.

System Action: The command ends.

User Response: Define the function, or set the **FPATH** variable to include a directory that contains a file with the command name. The contents of this file must include a function definition for the command name.

FSUMA924 !could not fork()

Explanation: **xargs** could not fork a new process. A system error indicating the cause is displayed with this message.

System Action: The program ends.

User Response: Take appropriate action based on the reason code.

FSUMA925 *path***: command not found**

Explanation: The command *path* cannot be found, or you do not have permission to access it.

System Action: The program ends.

User Response: Verify that you spelled the command correctly and that you have the correct permission to access it.

FSUMA932 *argument***: input file can't be a directory**

Explanation: Directories may not be used as input source.

System Action: The program ends.

User Response: Specify a non-directory file as input source.

FSUMA934 Warning: increment *number* **changed to** *number*

Explanation: The increment specified on the **nice** command resulted in a nice value that was not valid. The increment was adjusted to result in a valid nice value.

System Action: The command continues.

User Response: No action is needed. To avoid this warning, specify a priority increment that, when added to the current nice value, results in a valid nice value (0-39).

FSUMA935 *command* **not executable**

Explanation: The command specified as an argument on the **nice** command could not be run. A specific error description follows this message.

System Action: The **nice** command ends.

User Response: Specify an executable command as an argument on the **nice** command.

FSUMA957 translation code *code* **not valid unsigned char**

Explanation: The value of *code* fell outside the range of values for an unsigned character.

System Action: The program continues.

User Response: Change the variable type or the value assigned to it.

FSUMA958 duplicate translation on *character*

Explanation: During the processing of a new translation table, a character was redeclared.

System Action: The program continues.

User Response: Remove the extraneous declaration.

FSUMA959 undefined start condition *name*

Explanation: A start state was used in a pattern, but **lex** was unable to find it in the list of declared start states.

System Action: The program continues.

User Response: Declare the start state, or correct the name if it is misspelled.

FSUMA960 bad {NAME} syntax

Explanation: **lex** encountered a character in a `{NAME}` that was not an alphabetic, numeric, or _ (underscore). There could be a missing **}** at the end of a previous declaration.

System Action: The program continues.

User Response: Remove the illegal character(s) from the `{NAME}` declaration, or add the missing **}**.

FSUMA961 *definition* **not defined**

Explanation: After seeing a `{definition}`, **lex** was unable to find it in the list of declared substitutions.

System Action: The program continues.

User Response: Replace the declared substitution, or define it in the definition section.

FSUMA962 nested {*definition***} expansion**

Explanation: You may have attempted to place a call to a macro with the definition of that macro.

System Action: The program continues.

User Response: Remove the recursive macro call, or change the macro to a function that can be called recursively.

FSUMA963 ?possibly non-portable character class

Explanation: The defined character class may not be fully portable because of the inclusion of system or locale-specific characters, numbers, or symbols.

System Action: The program continues.

User Response: If you want to find a more portable (POSIX-defined) definition of the desired character class, consult *OS/390 UNIX System Services Command Reference* or a **lex** manual.

FSUMA964 Incomplete %{ declaration

Explanation: **lex** could not find the trailing **%}**, which needs to be the first and only thing on a line.

System Action: The program ends.

User Response: Add or correct the missing **%}**.

FSUMA965 Too many start conditions

Explanation: During processing of the definition section, the number of start conditions exceeded the size of **lex**'s static internal table. The default number of start conditions is 100.

System Action: The program ends.

User Response: Use fewer start conditions.

FSUMA966 can't determine look ahead

Explanation: **lex** ran into a conflict in lookahead tokens, and did not know how to resolve it.

System Action: The program continues.

User Response: Simplify or expand the grammar rules to remove the conflict.

FSUMB041 Mode *mode* not supported in this window

Explanation: The system was unable to set the number of rows in this window.

System Action: The program ends.

FSUMB082 result of substitution too long

Explanation: The resulting filename after the substitution specified on the **-s** option exceeds the maximum filename length allowed.

System Action: The program continues.

User Response: Use a shortened substitution string.

FSUMB091 literal newline characters are not allowed in EREs

Explanation: An attempt was made to place a newline character after the \.

System Action: The program continues.

User Response: Remove the literal newline from the extended regular expression.

FSUMF018 Bad byte count on ESD card: *offset*

Explanation: The object deck was corrupted.

System Action: The program ends.

User Response: Recompile the source to create another object deck.

FSUMF019 Bad length on XSD card: *offset*, *length*

Explanation: The object deck was corrupted.

System Action: The program ends.

User Response: Recompile the source to create another object deck.

FSUMF020 File *filename***: Unexpected end of file or archive member**

Explanation: The file or archive member does not end with an **end** record. The file or archive member was corrupted.

System Action: The program ends.

User Response: Reprocess the file or archive member. You may need to recompile the specified file.

FSUMF060 man: glob() failure

Explanation: The **glob()** function failed to generate a pathname. Possible reasons include:

- Insufficient permission to fully search directory paths
- Inability to allocate memory
- An internal error that caused the function to end

System Action: The program ends.

User Response: The pathname that was attempted is printed if the **-x** option was selected. It may be necessary to contact your system administrator if permissions are not set to allow access. If insufficient memory or storage was the cause, try freeing up system resources and reattempt the program.

FSUMF062 man: mkdir(*path*) failure

Explanation: The **mkdir()** function failed to create the named directory. Possible reasons for the failure include:

- The pathname names a symbolic link
- The process did not have search permission on some component of the pathname
- The pathname is too long (it exceeds **PATH_MAX**)
- There is insufficient storage
- A component of the pathname prefix is not a directory
- The parent directory is on a read-only file system

System Action: The program ends.

User Response: Check the pathname for any of the above errors. It may be necessary to contact your system administrator for errors involving permissions. In the event of insufficient storage, try freeing up system resources or adding more storage.

FSUMF159 *command*: **localtime() failed**

Explanation: A system call to convert the calendar to local time has failed.

User Response: Contact your system programmer.

System Programmer Response: There may be a problem with the **TZ** environment variable or the current **LC_TOD** locale category (assuming a POSIX environment).

FSUMF170 getsyntax: Cannot retrieve the variant character definitions from the current locale.

Explanation: The system was unable to locate variant character information and/or definitions for the current locale.

User Response: Contact the system programmer.

System Programmer Response: Ensure that the current locale has **LC_SYNTAX** defined correctly.

Part 2. APARs OW42811 and OW42841: OS/390 MVS Library

Chapter 5. APARs OW42811 and OW42841: OS/390 MVS System Messages

5.1 BPX Messages

BPXI038I TASK*procname* **HAS ABNORMALLY ENDED.** *text*

Explanation: The OS/390 UNIX task abnormally ended and cannot be recovered. The end of task exit routine (ETXR) failed to reattach it after a preset number of attempts.

text is one of the following:

MEMORY MAP PROCESSING IS SUSPENDED UNTIL THE NEXT IPL
OS/390 UNIX memory map processing is being suspended until the next IPL.

MODIFY BPXOINIT PROCESSING IS SUSPENDED
OS/390 UNIX MODIFY BPXOINIT console commands are being suspended until the next IPL.

NETWORK DISPATCHER WORKLOAD BALANCING IS SUSPENDED
The OS/390 UNIX Network Dispatcher workload balancing function is being suspended until the next IPL.

In the message text:

procname
The name of the OS/390 UNIX task that abnormally ended

Source: OS/390 UNIX System Services kernel (BPX)

Module: BPXQETXR

System Action: The system continues.

Operator Response: None.

System Programmer Response: The identified OS/390 UNIX task has ended. The function is unavailable until the next IPL. The system should have presented other information that identifies the cause of the task failure.

Chapter 6. APARs OW42811 and OW42841: OS/390 MVS Routing and Descriptor Codes

6.1 BPX Messages

Message Identifier	Routing Code	Descriptor Code
BPXB001E	1,10	3
BPXB002E	1	11
BPXB003I	2	4
BPXB004E	1	11
BPXB005I	2	4
BPXC001I	2	4
BPXF001I	2	4
BPXF002I	2	4
BPXF003I	2	4
BPXF004I	2,10	4
BPXF005I	2,10	4
BPXF006I	2	4
BPXF007I	2,10	4
BPXF008I	2,10	4
BPXF009I	2,10	4
BPXF010I	2,10	4
BPXF011I	2,10	4
BPXF012I	2,10	4
BPXF013I	2	4
BPXF014D	2	2
BPXF015I	*	5
BPXF016I	2	4
BPXF017I	2	4
BPXF018I	2	4
BPXF019I	2	4
BPXF020I	2	11
BPXF021I	2	4
BPXF022I	2	4
BPXF023I	2,10	4
BPXF024I	2	4
BPXF025I	2	4
BPXF026I	2	4
BPXF027I	2	4
BPXF028I	2	4
BPXF029E	2	11
BPXF030I	2,10	4
BPXF031I	2,10	4
BPXF032D	2	2
BPXF033I	2,10	4
BPXF101E	-	5
BPXF102E	2	5
BPXF103E	2	5
BPXF104E	2	5
BPXF105E	2	2
BPXF106E	2	2
BPXF107E	2	5
BPXF108E	2	5
BPXF110E	2	2
BPXF111E	2	2

Message Identifier	Routing Code	Descriptor Code
BPXF112W	2	2
BPXF113W	2	2
BPXF114E	-	5
BPXF115E	2	5
BPXF116E	-	5
BPXF117E	2	2
BPXF118W	2	2
BPXF119W	2	2
BPXF120E	2	5
BPXF121E	2	5
BPXF123E	2	2
BPXF124E	2	2
BPXF125E	2	2
BPXF126E	2	5
BPXF127E	2	5
BPXF128E	2	5
BPXF129E	2	5
BPXF130E	2	5
BPXF131E	2	5
BPXF132E	2	5
BPXF134E	2	2
BPXF135E	2	2
BPXF136E	2	5
BPXF137E	2	2
BPXF138E	2	2
BPXF139E	2	2
BPXF140E	2	2
BPXF141E	2	2
BPXF142E	2	2
BPXF143E	2	2
BPXF144I	-	-
BPXF145E	2	2
BPXF146E	2	2
BPXF147E	2	2
BPXF148E	2	2
BPXF150I	2	5
BPXF151I	2	5
BPXF152W	2	2
BPXF153W	2	2
BPXF154E	2	2
BPXF155E	2	2
BPXF156E	2	2
BPXF157E	2	2
BPXF158E	2	2
BPXF159E	2	5
BPXF160E	2	2
BPXF161I	2	2
BPXF162E	2	2
BPXF163E	2	2
BPXF164E	2	2
BPXF165E	2	2
BPXF166E	2	2
BPXF167E	2	2
BPXF168E	2	2
BPXF169E	2	2
BPXF170E	2	2
BPXF171E	2	2
BPXF172E	2	2

Message Identifier	Routing Code	Descriptor Code
BPXF173E	2	2
BPXF174E	2	2
BPXF175E	2	2
BPXF176E	2	2
BPXF201I	2,10	4
BPXF202I	2	4
BPXF203I	2	4
BPXF204I	2	4
BPXF205I	2	4
BPXF206I	2	4
BPXF207I	2	4
BPXF208I	2	4
BPXF209I	2	4
BPXF210I	2	4
BPXF211I	2,10	4
BPXF212I	2,10	4
BPXF213E	1, 2	3
BPXF214E	2	11
BPXF215E	2	11
BPXF216E	1, 2	3
BPXF217E	1, 2	3
BPXF218I	2	4
BPXI002I	2	4
BPXI003I	2	4
BPXI004I	2	4
BPXI005I	2	4
BPXI006I	-	4
BPXI007I	-	4
BPXI008I	-	4
BPXI009I	-	4
BPXI010I	-	4
BPXI011I	-	4
BPXI012I	2,10	4
BPXI013I	2,10	4
BPXI014I	2,10	4
BPXI015I	2	4
BPXI016I	2	4
BPXI017I	2	4
BPXI018I	2	4
BPXI019I	2	4
BPXI020I	2	4
BPXI021I	2	4
BPXI022I	-	4
BPXI023I	-	4
BPXI024I	-	4
BPXI025I	-	4
BPXI026I	2	4
BPXI027I	2	4
BPXI028E	1	11
BPXI029I	1,2,10	12
BPXI030I	1,2,10	12
BPXI031E	1	1
BPXI032E	1,10	11
BPXI033E	1,10	11
BPXI034I	2	4
BPXI035E	1	11
BPXI038I	2	4
BPXM001I	11	6

Message Identifier	Routing Code	Descriptor Code
BPXM002I	11	6
BPXM004I	11	6
BPXM006I	11	6
BPXM007I	11	6
BPXM008I	11	6
BPXM009I	11	6
BPXM010I	11	6
BPXM011I	11	6
BPXM012I	11	6
BPXM013I	11	6
BPXM014I	11	6
BPXM015I	11	6
BPXM016I	11	6
BPXM017I	11	6
BPXM018I	11	6
BPXM019I	11	6
BPXM020I	11	6
BPXM021E	2	5
BPXM022E	2	5
BPXM023I	2	4
BPXM024I	2	4
BPXM025I	2	4
BPXM026I	2	4
BPXM027I	2	4
BPXM028I	2	4
BPXM029I	2	4
BPXM030I	2	12
BPXM031I	2	12
BPXM032E	1,10	11
BPXM033I	2	12
BPXM036I	2	4
BPXM037I	2	4
BPXM038I	2	4
BPXM039I	2	4
BPXM040I	2	4
BPXM041I	2	4
BPXM042I	2	4
BPXM043I	2	4
BPXM047I	11	6
BPXN001I	2	4
BPXN002I	2	4
BPXO001I	#	5,8,9
BPXO002I	#	5,8,9
BPXO003I	#	5,8,9
BPXO006I	2	5
BPXO007I	2	5
BPXO008I	2	5
BPXO009I	2	5
BPXO012I	2	5
BPXO015I	2	5
BPXO016I	2	5
BPXO017I	2,10	4
BPXO024I	2	5
BPXO025I	2	5
BPXO026I	2	5
BPXO027I	2	5
BPXO028I	2	5
BPXO029I	2	4

Message Identifier	Routing Code	Descriptor Code
BPXO030I	2	5
BPXO031I	2,10	4
BPXO032I	2	5
BPXO033I	2,10	4
BPXO034I	2	5
BPXO035I	2,10	4
BPXO036I	2	5
BPXO037E	2	5
BPXO038I	2	5
BPXO039I	2, 10	4
BPXO040I	-	5,8,9
BPXO041I	-	5,8,9
BPXO042I	-	5,8,9
BPXO043I	-	5,8,9
BPXO044I	-	5,8,9
BPXO045I	-	5,8,9
BPXO046I	-	5,8,9
BPXO047I	-	5,8,9
BPXO048I	2	5
BPXP001I	2	4
BPXP003E	1,10	11
BPXP004E	1,10	11
BPXP005I	-	4
BPXP006E	1,10	11
BPXP007E	1,10	11
BPXP008E	1,10	11
BPXT001I	2,10	4
BPXU001I	2	4
BPXU002I	2	4
BPXU003I	2	4
BPXU004I	2	4
BPXU005I	2,10	4
BPXW0000I	2	2
BPXW0001I	2	2
BPXW0002I	2	2
BPXW0003I	2	2
BPXW0004I	2	2

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Mail Station P300
2455 South Road
Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming Interface Information

This book is intended to help the customer plan for, customize, operate, manage, and maintain an OS/390 system with OS/390 UNIX System Services (OS/390 UNIX).

This book primarily documents intended Programming Interfaces that allow the customer to write programs that use OS/390 UNIX.

This book also documents information that is NOT intended to be used as Programming Interfaces of OS/390 UNIX. This information is identified where it occurs, either by an introductory statement to a chapter or section or by the following marking:

NOT Programming Interface information

End of NOT Programming Interface information

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

AnyNet
CICS
CICS/ESA
DFSMS/MVS
DFSMSdfp
DFSMShsm
IBM
IMS
Language Environment
OS/390
RACF
RMF
VTAM

Lotus, Domino, and Lotus Go Webserver are trademarks of the Lotus Development Corporation.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others:

DFS	Transarc Corporation

Index

Communicating Your Comments to IBM

OS/390
UNIX System Services: APARs OW42811 and OW42841

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply.

Feel free to comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. However, the comments you send should pertain to only the information in this manual and the way in which the information is presented. To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

If you are mailing a reader's comment form (RCF) from a country other than the United States, you can give the RCF to the local IBM branch office or IBM representative for postage-paid mailing.

- If you prefer to send comments by mail, use the RCF at the back of this book.
- If you prefer to send comments by FAX, use this number:
  - FAX: (International Access Code)+1+914+432-9405
- If you prefer to send comments electronically, use one of these network IDs:
  - IBM Mail Exchange: USIB6TC9 at IBMMAIL
  - Internet e-mail: mhvrcfs@us.ibm.com
  - World Wide Web: http://www.ibm.com/s390/os390/

Make sure to include the following in your note:

- Title and publication number of this book
- Page number or topic to which your comment applies

Optionally, if you include your telephone number, we will be able to respond to your comments by phone.

Reader's Comments — We'd Like to Hear from You

OS/390
UNIX System Services: APARs OW42811 and OW42841

You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you. Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

Note: Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

Today's date: ______________________________

What is your occupation?

Newsletter number of latest Technical Newsletter (if any) concerning this publication:

How did you use this publication?

[ ] As an introduction
[ ] As a reference manual
[ ] For another purpose (explain)

[ ] As a text (student)
[ ] As a text (instructor)

____

____

Is there anything you especially like or dislike about the organization, presentation, or writing in this manual? Helpful comments include general usefulness of the book; possible additions, deletions, and clarifications; specific errors and omissions.

Page Number: Comment:

____
Name

____
Company or Organization

____
Phone No.

____
Address

____

____

Reader's Comments — We'd Like to Hear from You

Cut or Fold
Along Line

IBM®

Fold and Tape

Please do not staple

Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL   PERMIT NO. 40   ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Department 55JA, Mail Station P384
2455 South Road
Poughkeepsie, NY 12601-5400

Fold and Tape

Please do not staple

Fold and Tape

Cut or Fold
Along Line

Program Number: 5647-A01

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.